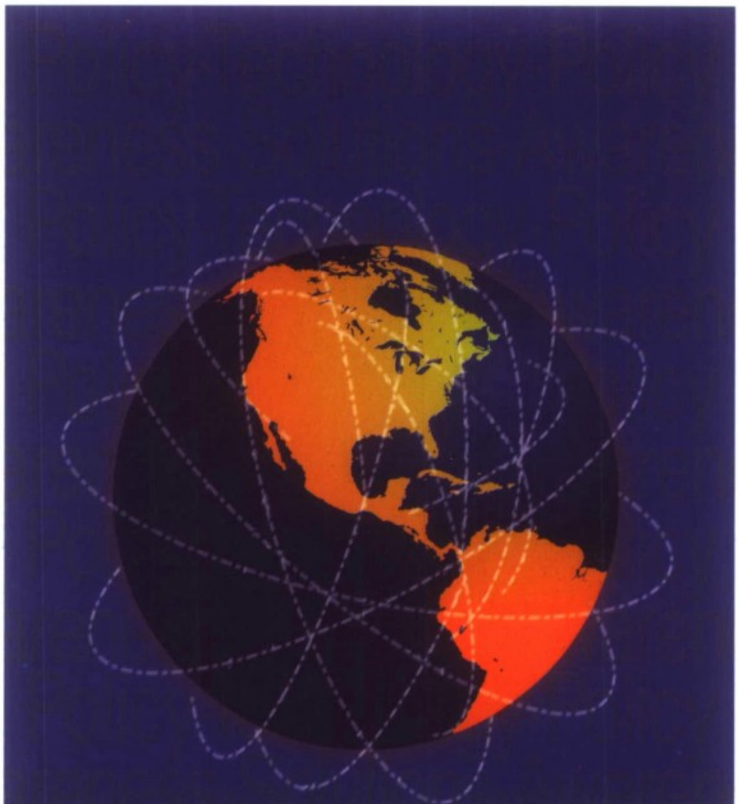
19th

National Information Systems Security Conference

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY

NATIONAL COMPUTER SECURITY CENTER



NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY

NATIONAL COMPUTER SECURITY CENTER

October 22–25, 1996
Baltimore Convention Center
Baltimore, MD

Volume 2

DTIC® has determined on 04/10/2009 that this Technical Document has the Distribution Statement checked below. The current distribution for this document can be found in the DTIC® Technical Report Database.

☒ **DISTRIBUTION STATEMENT A.** Approved for public release; distribution is unlimited.

☐ **© COPYRIGHTED**; U.S. Government or Federal Rights License. All other rights and uses except those permitted by copyright law are reserved by the copyright owner.

☐ **DISTRIBUTION STATEMENT B.** Distribution authorized to U.S. Government agencies only (fill in reason) (date of determination). Other requests for this document shall be referred to (insert controlling DoD office)

☐ **DISTRIBUTION STATEMENT C.** Distribution authorized to U.S. Government Agencies and their contractors (fill in reason) (date of determination). Other requests for this document shall be referred to (insert controlling DoD office)

☐ **DISTRIBUTION STATEMENT D.** Distribution authorized to the Department of Defense and U.S. DoD contractors only (fill in reason) (date of determination). Other requests shall be referred to (insert controlling DoD office).

☐ **DISTRIBUTION STATEMENT E.** Distribution authorized to DoD Components only (fill in reason) (date of determination). Other requests shall be referred to (insert controlling DoD office).

☐ **DISTRIBUTION STATEMENT F.** Further dissemination only as directed by (inserting controlling DoD office) (date of determination) or higher DoD authority.

Distribution Statement F is also used when a document does not contain a distribution statement and no distribution statement can be determined.

☐ **DISTRIBUTION STATEMENT X.** Distribution authorized to U.S. Government Agencies and private individuals or enterprises eligible to obtain export-controlled technical data in accordance with DoDD 5230.25; (date of determination). DoD Controlling Office is (insert controlling DoD office).

Welcome

The National Computer Security Center (NCSC) and the National Institute of Standards and Technology are pleased to welcome you to the Nineteenth National Information Systems Security Conference. We believe the conference will stimulate a productive information exchange and promote a greater understanding of today's information security issues and protection strategies.

The conference program addresses a wide range of interests from technical research and development projects to user-oriented management and administration topics. In today's ever more complex world where competitiveness demands swift, secure, value-added solutions, industry and government security professionals need to know how their vital information systems are threatened, what the vulnerabilities are, and how they can implement solutions. This Conference provides a unique international forum covering a wide variety of information systems security issues. Papers and panels in this multitrack program cover security issues related to: the Internet, electronic commerce, firewalls, information warfare, legal issues, computer crime, the World Wide Web, incident handling, cryptography, viruses, research and development, policies, vulnerabilities and threat, assurance, security engineering, and much more. As our technology increases, more enterprises are recognizing their need for computer security. The special sessions on electronic commerce and legal issues should be of particular interest to organizations that are starting to do business electronically.

The vendor exposition, sponsored by the Armed Forces Communications and Electronics Association (AFCEA) and held in parallel with this Conference, provides a forum for industry to showcase information systems security technology and provides hands-on demonstration of products and services that are potential solutions to many network and computer security problems.

We believe that the professional contacts you make at this conference, the presentations, and these Proceedings will offer you insights and ideas you can apply to your own security planning efforts. We encourage you to share the ideas and information you acquire this week with your peers, your management, and your customers. We also encourage you to share with us your successful security techniques as well as your thoughts and discussions about the problems you are experiencing and anticipate. It is through this exchange that we will continue to enhance the security of our information systems and networks and build a strong foundation to make security a credible value-added part of your enterprise such that security, policy, and technology truly are partners in your enterprise.

Shukri Wakid

SHUKRI A. WAKID
Director
Computer Systems Laboratory

JOHN C. DAVIS
Director
National Computer Security Center

This Page

Intentionally

Left Blank

RISE OF THE MOBILE STATE:

ORGANIZED CRIME IN THE 21ST CENTURY

By

August Bequai, ESQ.
McLean, VA 22102

National Information Systems
Security Conference
October 22, 1996
Baltimore, MD

An associate of a New York Mafia family, is alleged to have orchestrated a multimillion dollar theft of microchips from a West Coast firm. A member of a European crime syndicate is said to have created fictitious accounts on the computers of a bank, and then used the funds to purchase securities. Members of an Asian crime family are said to have used the E-mail system of a multinational financial institution, to launder monies from their illegal operations.

Organized crime is a growth industry both within and outside the U.S. The fragmented global political environment has served to abet its growth. In the U.S. alone, organized crime is said to gross more than $200 billion annually. No nation is immune from its tentacles. Security experts fear that the international crime syndicates are, increasingly, going high-tech. In large part, capitalizing on the implements of the IT revolution.

Asian, European, African, and Latin American crime syndicates are joining forces and pooling their resources; becoming a political and economic power in the global scene - a "mobile state", that rivals the multinational corporate giants in political and economic clout. Like the multinationals, the crime syndicates operate free of national restraints; guided by economic motives. In the process, they have harnessed the IT revolution.

Organized crime has learned to subvert IT so as to enhance its predatory practices; as well as augment its power and evade prosecution. Like the nomadic tribes of antiquity, who used the mobility of their fast steeds to prey on organized societies, these criminal mobile states are learning to implement EDI, the Internet, and other IT vehicles to their ends.

Why the Threat

Well into the 1980s, the international community, dismissed the threat of the global crime syndicates as the creation of Hollywood; while it made for good entertainment, it was not taken seriously. Even the high-tech security establishment, fixated with hackers, focused little or no attention on the threat posed by the crime cartels. The IT security literature of the 1990s, replete with stories of cyber-crime and hackers, is noticeably devoid of any mention of organized crime; even a tangential one. The threat of syndicated crime in the IT environment, has been sublimated; nor have any efforts been made to study it.

The international crime syndicates have, historically, demonstrated an uncanny ability to employ the tools of technology in their arsenal. They have learned to adapt to their environment. The U.S. syndicates, and not the banks, made first extensive use of the wire services in the 1930s. The U.S. syndicates also employed, with success, the telephone, radio, air travel, and other technologies, to expand their operations over vast areas of North America. The growth of the U.S. Mafia in the 1930s can, in large part, be attributed to new technologies of that period. Its multibillion dollar gambling empire would not have been possible without the rise of telephonic communications. The Internet, should likewise, serve them well.

The crime syndicates have also demonstrated an ability to subvert both business and government. Blackmail, extortion and the threat of potential violence have been employed with noticeable success. In Italy, organized crime has even been able to topple governments; in Asia, the Triads and Yakuza helped their political allies gain political ascendancy. In Latin America, they have battled governments and left leaving revolutionary movements with success. They have demonstrated both the will and means to both survive and prevail.

But unfortunately, the international community has both neglected and underestimated the ability of the crime syndicates to employ the tools of IT in their illicit operations. While state-sponsored terrorism and the antics of religious zealots capture the daily headlines, the multibillion dollar EFT transactions of the drug cartel go unnoticed.

While modern terrorists constitute a growing problem, the ability and willingness of the crime cartels to terrorize and cause havoc, should not be dismissed. The Columbian syndicates have long since laid such doubts to rest.

But organized crime, even more so than the modern terrorists, is attuned to subtle vulnerabilities of the body politic of the nation-state. For example -

(1) The crime syndicates have been known to extort monies from businesses and governments, in return for security. For example, the Asian syndicates were successful in keeping the extreme Left at bay, in return for political favors; in Italy, the Mafia decimated the Sicilian Communist party, in return for immunity from prosecution.

(2) The syndicates have had little difficulty in coercing bankers to assist them in their money-laundering activities; or to tap into the multibillion dollar pension funds of labor unions.

(3) The syndicates have been known to join forces with political radicals, when it meets their needs; as well as severing those alliances when their needs dictate otherwise. Asia and Latin are replete with examples of the drug cartels establishing alliances of convenience.

(4) While the power base of the crime cartels is not based on geography, as is the case with the nation-state, they will exert control over defined territory when necessary. For example, the now defunct state of Herzeg Bosna served for a short period of time, as a haven for Balkan crime syndicates.

Exploiting the IT Revolution

While the IT revolution has amply demonstrated its worth, unfortunately, the environment in which it operates, is far from idyllic. The potential for criminal abuse is very real. Transnational crime syndicates operate with impunity in the current environment; the international organizations that were established to curtail their activities, have failed to do so. The syndicates not only prey on the user community; but they have also learned to employ the implements of IT to expand and enhance their control over their expanding illicit operations. EFT and related electronic payment systems, have dramatically facilitated the transborder movement of syndicate money.

Structure of the Syndicate

The very term syndicate or organized crime - these are frequently used interchangeably in the U.S., to denote organized criminal activity, as opposed to traditional street crime- evokes images of a handful of poorly educated individuals; from the lower strata of society, who meet secretly in dingy smoke-filled basements. Over the years, numerous efforts have been made in the U.S. and Europe to study and analyze the crime syndicates; the focus, however, has been on the European and U.S. Mafia groups. The Asian syndicates have largely escaped scrutiny. Hollywood continues to portray these groups as monoliths; dominated by chieftains of Mediterranean descent.

But organized crime is much more complex; as well as international in its operations. Crime syndicates permeate the societal fiber of every country. Some have their roots in Medieval History; evolving and adapting over the centuries. They go by different names - i.e., Yakuza, Triad, Camora, Mafia, Unione Corse, etc. - and exhibit diverse traits and modes of behavior. Some of them are historical rivals. But most of them share certain commonalities; among these -

(1) Their basic structure and organization is largely feudal and highly decentralized; resembling the tribes and clans of the Medieval world, rather than the modern organizations that they prey on. Had they been monoliths, they would have proven easy to decapitate.

(2) Their primary loyalty lies not with the nation-states from which they operate, but rather to the organization to which they belong; as well as its leadership.

(3) Even the more sophisticated of the crime syndicates, idealizes the past; when civilization was less complex and simple. Post-industrial societies are viewed as decadent. The Yakuza, for example, look back fondly to the age of the Samurais; they view modern Japan with disdain.

(4) While the syndicates pay lip-service to the idyllic past, they are driven by economic motives; selling their services to the highest bidder. For example, the Lebanese syndicates, while paying lip-service to Islam, sell their services to Muslims and Christians alike.

(5) The syndicate families are bound together largely by kinship and blood ties. They often share a similar tradition and culture; as well as loyalty to the group. The nation-state and its laws, are merely tolerated.

(6) The international syndicates are mobile in nature; with associates in many geographic areas. For example, the Triad syndicates have associates in Asia, North America and Europe.

While the criminal syndicates of the Medieval period operated, within confined geographic areas - the result of limitations imposed on them by the primitive technologies of their era - those of the IT society, operate globally. They make widespread use of IT to communicate with each other; as well as free themselves of the constraints of the nation-state. The IT revolution has given them mobility.

The Turning Point

Secret criminal societies have been with us since the dawn of civilization. They are the antithesis to organized government. The early twentieth century witnessed the rise and proliferation of criminal syndicates around the world; their expansion was abetted, in part, by the new technologies resulting from the industrial revolution. The urbanization of modern societies added fuel to their growth.

The turning point for the international syndicates came in the post-World War II period. Until then, the crime cartels had been fragmented, regional, and limited in their operations to specified geographic areas. The post-World War II period witnessed the rise of new technologies and proliferation of new communication systems. Television became a household fixture. Armed with these technologies, the syndicates began to make their appearance on the global scene as powers to be reckoned with.

The new syndicate leadership, reared in the high-tech environment, turned its attention to international commerce. The syndicates embraced the world of high-technology; unfortunately, law enforcement failed to keep abreast. The modern syndicates must be viewed as a fusion of modern technology and a feudal organizational structure. This serves to make them dangerous to the post-industrial society; as well as impervious to its law enforcement apparatus.

Syndicates Embrace IT

IT lends itself to three key areas of syndicate activity: first, it makes the detection and prosecution of their illicit activities more difficult; secondly, it creates new targets of opportunity for them in the high-tech sector; and thirdly, it enhances their ability to coordinate and manage their global operations. With regard to the first, the failure of police agencies the world over to stay abreast of the IT revolution, has made the prosecution of the syndicates much more difficult.

Secondly, the IT revolution has opened new opportunities for the syndicates; i.e., computer/E-mail crimes, data thefts, computer sabotage, high-tech pornography, money laundering, and so forth. The third area, makes it possible for the syndicates to communicate by E-mail, EDI, and so on; it also serves to evidence their global mobility, and challenge the power of the nation-state.

High-Tech Crimes

IT has facilitated the commission of high-tech crimes by the syndicates. It can be employed to commit sophisticated wire frauds, commodity swindles, embezzlements, and other crimes. The multimillion dollar high-tech assisted swindles in the world of international finance, amply evidence the power of IT as a vehicle for the syndicates.

The syndicate have, over the years, been heavily involved in the financial frauds area. Syndicate controlled financial institutions, have been used in sophisticated high-tech frauds; as well as money laundering operations. The syndicate has also demonstrated an ability to employ IT in other endeavors. To cite a few examples -

- Data thefts
- Computer frauds and sabotage
- EFT crimes
- Bankruptcy frauds

- Insurance scams
- Securities swindles
- Real estate scams
- Industrial espionage
- Theft of pension funds
- Payoff and kickback schemes
- Trafficking in stolen property

The use of IT in frauds against the government has also proved inviting to the syndicates; for example -

- Diversion of government funds
- Government contract frauds
- Theft of confidential data
- Sabotage of information systems
- Tax frauds

The potential for misuse of IT by the syndicates is real and serious. The ability of the syndicates to prey on the post-industrial society has increased with the IT revolution. The latter has made it more difficult to secure the nation-state from syndicate attacks. The failure of the nation-state to develop the requisite tools to combat syndicate activities, has proven of help to the latter.

Going Cashless

The IT revolution has also prompted a revolution in the world of finance. Electronic payment systems now dominate international banking. Trillions of dollars are transferred by electronic means every hour. Efforts to secure these electronic systems from syndicate attack have fared ill.

Through the use of electronic banking systems, the syndicates can hide the billions of dollars that they collect from their drug trade and other illicit operations. IT has also provided the syndicates with necessary mobility to evade prosecution.

(1) Extra-territorial activities by nation, aimed directly at the syndicates and their allies.

(2) Mobile police forces, that can operate internationally.

(3) IT safeguards to vend-off syndicated activities.

Aggressive steps need to also be taken by businesses to deter the illicit activities of the international syndicate. First and foremost, they need to enact security measures aimed at safeguarding their own IT systems. These should include -

(1) Securing databases from unauthorized access, deletions, alterations and/or manipulation.

Combatting the Mobile State

Given their vast resources, the international crime syndicates pose a formidable challenge to the modern nation-state. Their mobility and transborder operations, hamper the traditional efforts of the nation-state to curtail their operations. Both international cooperation and programs are needed to deter and contain syndicate activities. These should include -

- International mobile police forces that can traverse frontiers.
- Treaties aimed at attacking the financial power bases of the syndicates.
- Training for law enforcement agencies, in the detection, investigation, and prosecution of syndicate IT crimes.
- Security measures for international networks, databases, EDI, E-mail, EFT, and related technologies.
- Enhanced security awareness for both private and public officials.
- Laws specifically directed at facilitating the prosecution of syndicate criminal activities.

Summary

The international crime syndicates are neither monoliths nor parochial in their operations. Asian syndicates have been known to work closely with their European and North American counterparts. While the various syndicates may differ in structure, organization, and motives, the IT revolution has accorded them new opportunities and enhanced mobility. They traverse the globe at-will; coordinating their efforts, in large part, through the vehicles of the IT revolution. Like the Mongols and other nomadic marauders of antiquity, they constitute mobile states. The IT revolution has given them a power base from whence they can threaten havoc to the nation-state; the latter must respond.

19th National Information Systems Security Conference

Co-Chairs

Stephen F. Barnett, *National Computer Security Center*
Tim Grance, *National Institute of Standards and Technology*

Program Directors

Ellen Flahavin, *National Institute of Standards and Technology*
Jack Holleran, *National Computer Security Center*

Program Committee

Edward Borodkin, *National Computer Security Center*
Christopher Bythewood, *National Computer Security Center*
Sally Meglarthary, *Estee Lauder*
Dr. Gary Smith, *Arca Systems*

Administration

Tammie Grice, *National Institute of Standards and Technology*
Mary Groh, *National Computer Security Center*
Kathy Kilmer, *National Institute of Standards and Technology*
C. A. O'Brien, *National Computer Security Center*
Melissa Petherbridge, *National Computer Security Center*
Phyllis Pierce, *National Computer Security Center*
Pat Purkey, *National Security Agency*
Sara Torrence, *National Institute of Standards and Technology*

Conference Referees

Dr. Marshall Abrams	*The MITRE Corporation*
Rowland Albert	*National Security Agency*
James P. Anderson	*J. P. Anderson Company*
Devolyn Arnold	*National Security Agency*
James Arnold	*National Security Agency*
Alfred Arsenault	*National Security Agency*
Dr. D. Elliott Bell	*Mitretek Corporation*
Dr. Matt Bishop	*University of California, Davis*
Earl Boebert	*Sandia National Laboratory*
Dr. Dennis Branstad	*Trusted Information Systems, Inc.*
Dr. Martha Branstad	*Trusted Information Systems, Inc.*
Dr. Blaine Burnham	*National Security Agency*
Christopher Bythewood	*National Computer SecurityCenter*
Dr. William Caelli	*Queensland University of Technology, Australia*
Dr. John R. Campbell	*National Security Agency*
Lisa Carnahan	*National Institute of Standards and Technology*
Dr. Jon David	*The Fortress*
Dr. Dorothy E. Denning	*Georgetown University*
Donna Dodson	*National Institute of Standards and Technology*

Conference Referees *(continued)*

Karen Ferraiolo	*Arca Systems*
Ellen Flahavin	*National Institute of Standards and Technology*
Dan Gambel	*General Research Corporation*
Virgil Gibson, CISSP	*Computer Sciences Corporation*
Dennis Gilbert	*National Institute of Standards and Technology*
Barbara Guttman	*National Institute of Standards and Technology*
Dr. Grace Hammonds	*AGCS, Inc.*
Cindy Hash	*National Security Agency*
Ronda Henning	*Harris Corporation*
Dr. Harold Highland, FICS, FACM	*Computers & Security*
Jack Holleran	*National Computer Security Center*
Hillary H. Hosmer	*Data Security*
Carole Jordan	*Grumman Data Systems*
Steve Kougoures	*National Security Agency*
David Krehnke	*Lockheed Martin Energy Systems*
Helmut Kurth	*Industricanlagen Betriebsghesellschaft mbH (IABG), Germany*
Carl Landwehr	*Naval Research Laboratory*
Robert Lau	*National Security Agency*
Dr. Theodore M. P. Lee	Independent Consultant
Special Agent John Lewis	*United States Secret Service*
Steven Lipner	*Trusted Information Systems, Inc.*
Joseph Lisi	*National Security Agency*
Teresa Lunt	*Defense Advanced Research Projects Agency*
Wayne Madsen	*Computer Sciences Corporation*
John McDermott	*J - K International Limited*
Dr. John McLean	*Naval Research Laboratory*
Sally Meglathery	*Estee Lauder*
William H. Murray	*Deloitte & Touche*
Ruth Nelson	*Information System Security*
Dr. Peter Neumann	*Stanford research Institute, International*
Dr. Charles Pfleeger	*Trusted Information Systems, Inc.*
W. Timothy Polk	*National Institute of Standards and Technology*
Marcus Ranum	*V-ONE*
Marvin Schaefer	*Arca Systems*
Dr. Gary Smith	*Arca Systems*
Dr. Eugene Spafford	*Coast Laboratory, Purdue University*
Julian Straw	*Syntegra, UK*
James Tippett	*National Security Agency*
Ken van Wyk	*Science Applications International Corporation*
John Wack	*National Institute of Standards and Technology*
Mark Wallace	*National Security Agency*
Howard Weiss	*SPARTA, Inc.*
Valerie Williams	*Data Sciences, UK*
Roy Wood	*National Security Agency*
Mark Woodcock	*National Security Agency*
Paul Woodie	*National Security Agency*
Thomas Zmudzinski	*Defense Information Systems Agency*

AWARDS CEREMONY

2:00 p.m. Thursday October 24

Baltimore Convention Center, Room 337-338

The National Institute of Standards and Technology (NIST) and the National Computer Security Center (NCSC) will honor those vendors who have successfully developed products meeting the standards of the respective organizations. Immediately following the ceremony, honored vendors will have the opportunity to display these products.

The NCSC recognizes vendors who contribute to the availability of trusted products and thus expand the range of solution from which customers may select to secure their data. The products are placed on the Evaluated Products List (EPL) following a successful evaluation against the Trusted Computer Systems Evaluation Criteria including its interpretations: Trusted Database Interpretation; Trusted Network Interpretation; and Trusted Subsystems Interpretation. Vendors who have completed the evaluation process will receive a formal certificate of completion from the Director, NCSC marking the addition to the EPL. Certificates will also be presented to those vendors that have placed a new release of a trusted product on the EPL by participation in the Ratings Maintenance Program (RAMP). Additionally, vendors will receive honorable mention for being in the final stages of an evaluation as evidenced by transition into the Formal Evaluation phase. The success of the Trusted Product Evaluation Program is made possible by the commitment of the vendor community.

The Computer Security Division at NIST provides validation services to test vendor implementations for conformance to security standards. NIST currently maintains validation services for three Federal Information Processing Standards (FIPS): FIPS 46-2, Data Encryption Standards (DES); FIPS 113, Computer Data Authentication; and FIPS 171, Key Management Using ANSI X9.17. During this award ceremony, NIST presents "Certificate of Appreciation" awards to those vendors who have successfully validated their implementation of these standards.

With the reaffirmation of the Data Encryption Standard as FIPS 46-2 in 1993, DES can now be implemented in software, as well as hardware and firmware. To successfully validate an implementation for conformance to FIPS 46-2, a vendor must run the Monte Carlo test as described in NBS (NIST) Special Publication 500-20. The Monte Carlo test consists of performing eight million encryptions and four million decryptions, with two encryptions and one decryption making a single test.

Vendors test their implementations of conformance to FIPS 113 and its American National Standards Institute (ANSI) counterpart, ANSI X9.9, Financial Institution Message Authentication (Wholesale). This is done using an electronic bulletin board system. Interactive validation requirements are specified in NBS (NIST) Special Publication 500-156, Message Authentication Code (MAC) Validation System: Requirements and Procedures. The test suite is composed of a series of challenges and responses in which the vendor is requested to either compute or verify a MAC on given data using a specified key which was randomly generated.

Conformance to FIPS 171 is also tested using an interactive electronic bulletin board testing suite. FIPS 171 adopts ANSI X9.17, Financial Institution Key Management (Wholesale). ANSI X9.17 is a key management standard for DES-based applications. The tests are defined in a document entitled NIST Key Management Validation System Point-to-Point (PTP) Requirements. The test suite consists of a sequence of scenarios in which protocol messages are exchanged under specified conditions.

We congratulate all who have earned these awards.

19th National Information Systems Security Conference

Refereed Papers

Criteria & Assurance — Track A

Electronic Commerce — Track B

Legal Perspectives — Track E

Management & Administration — Track F

Research & Development — Track G

Solutions — Track H

Invited Papers

Management & Administration — Track F

Legal Perspectives — Track E

Panels

Criteria & Assurance — Track A

Electronic Commerce — Track B

In Depth **Track C**

Management & Administration — Track F

Research & Development — Track G

Solutions — Track H

Closing Plenary Session

19th National Information Systems Security Conference

Author Cross Reference Index

19th National Information Systems Security Conference

Author Cross Reference Index

19th National Information Systems Security Conference

Author Cross Reference Index

SECURITY POLICIES FOR THE FEDERAL PUBLIC KEY INFRASTRUCTURE

Noel A. Nazario
NIST North, Room 426
820 West Diamond Avenue
Gaithersburg, MD 20899
NNazario@nist.gov

Introduction and Background

This paper discusses provisions for the handling of security policies in the proposed Federal Public Key Infrastructure (PKI). As shown in Figure 1, the proposed Federal PKI [1] is a public key certificate management system organized administratively as a hierarchy of Certification Authorities (CAs), and their Organizational Registration Authorities (ORAs), that rely on a Directory Service (DS) [2] to disseminate certificates and Certificate Revocation Lists (CRLs). The certificates managed by the PKI [4] support widespread use of digital signatures, and other public key enabled security services, by binding public keys to individuals, roles, or processes and allowing the verification of the authenticity of digital signatures. CAs certify PKI users and each other (cross-certification) to establish trust relationships and define both hierarchical and networked verification paths for user certificates. Hierarchical paths are established by

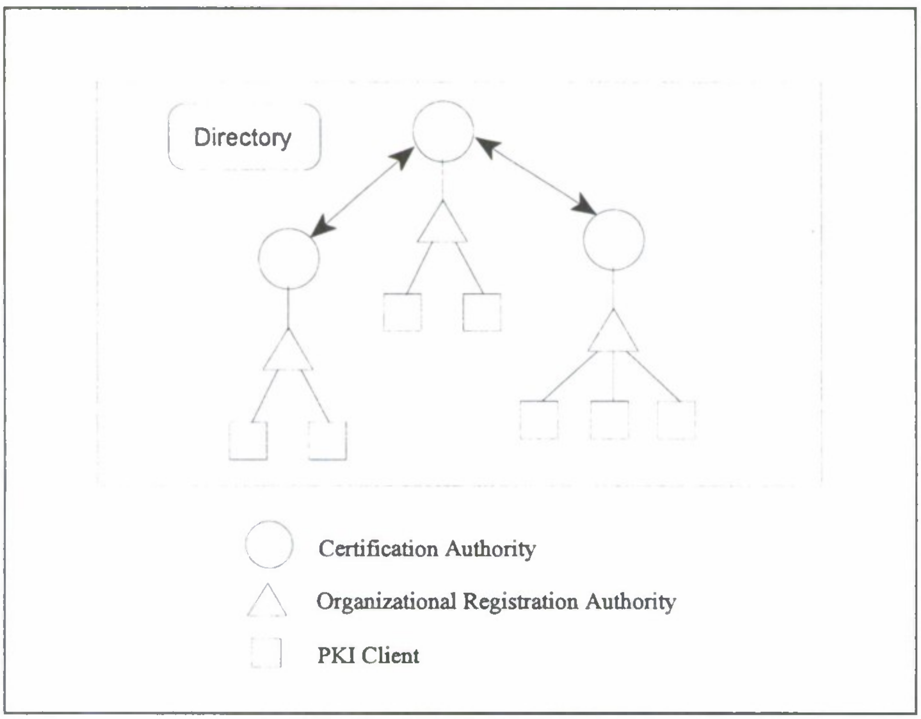


Figure 1 - Main Components of Federal PKI

following the certificate path from a root CA to the originator, networked paths are established by finding the appropriate cross-certificates connecting the CAs between the originator and the verifier. Trust is delegated hierarchically and most cross-certificates are required to preserve that delegation. CAs also certify ORAs that verify the identity of users and then vouch them to the CA when requesting initial certification. CA certificates are obtained by one or more agents (authorized operators) of the CA on behalf of the CA, not of the agent(s). ORA certificates are obtained by the agents on their own behalf, i.e., ORA signatures are bound to the agent, not to the ORA.

The Trusted Computer System Evaluation Criteria (TCSEC) [3] defines security policy as a "set of laws, rules, and practices that regulate how an organization manages, protects, and distributes sensitive information." Federal PKI policies deal with the generation, revocation, and dissemination of public key certificates, the integrity of the infrastructure, maintenance of records, identification of certificate holders, and the establishment of trust relationships between CAs. The verification of a digital signature is not sufficient indication of the trustworthiness of an electronic message or data file. The verifier needs to factor the trustworthiness of the CAs involved in the certification of the signatory. This is accomplished by examining the certificate policies for those CAs. Federal PKI certificates include a certificate policy field that identifies the security policy under which the certificate was issued. To enable a reasonable judgement on whether to accept a signed document or message, the certificate policy field of the corresponding certificate should point to information about the certificate issuing rules and about the trustworthiness of the CA that granted the certificate. The strictest certificate issuance rules are meaningless if the system that grants the certificate does not verify and protect the integrity of the certificates it generates and does not handle archiving, posting, and revocation of certificates responsibly. The Federal PKI Technical Security Policy (TSP) [5] defines CA Operational Policies and Certificate Issuance Policies that combine into certificate policies that are conveyed by the certificates. CA Operational Policies define the operation of CAs; Certificate Issuance Policies state identification requirements for parties requesting certification.

To ease the assessment of the trustworthiness of a certificate, the TSP defines three Federal Assurance Levels (low, medium, and high). These assurance levels are assigned to each CA by a Policy Approving Authority (PAA) that reviews its policies and practices and determines the highest assurance level that the CA can assign to the certificates it creates. Although they are not actual policies, the identifier for any of these assurance levels can be included in the certificate policies field and used when deciding whether to trust the certificate and the signed document verified with the public key in it. The Federal Assurance Levels are understood by all CAs in the proposed Federal PKI. The PAA performs periodic reviews of the operations of CAs to ensure that an even level of service is maintained throughout the infrastructure.

The policy guidelines discussed in this document apply to all components of the proposed Federal PKI, including CAs, ORAs, directory servers, et cetera. Federal PKI policies are enforced by PAAs.

CA Operational Policy

A CA Operational Policy explicitly defines the operation of a CA. This includes: backup procedures, record archiving procedures, qualifications of operations personnel, functional roles of CA operators, physical protection of the CA, Federal Information Processing Standard (FIPS) 140-1 security level requirements for CA cryptographic modules [6], access controls for CA private keys, et cetera. The CA Operational Policies of CAs in the Federal PKI will be posted in the NIST Computer Security Objects Register (CSOR) [7].

The TSP defines minimum operational requirements for all Federal CAs and additional assurance level-specific requirements for three Federal Assurance Levels; low, medium, and high. The requirements for each level of assurance include the level-specific requirements in addition to those for the level below it.

Minimum Operational Requirements

All CAs within the Federal PKI sign certificates using FIPS-approved signature algorithms. CAs may either generate any parameters that may be required by their signature algorithm or obtain them from the parent CA. The security policy determines the source of such parameters. The validity period of these parameters is established by the policy of the CA that defines them. The parameters will be maintained for the specified period unless the system is compromised and/or corruption of the locally-maintained certificate-generation data occurs. The parent CA may also request that the parameters used by its subordinates be changed if it is compromised or its database becomes corrupted. If any are required, algorithm parameters will be included in the Subject Public Key Field of the every certificate.

All CAs perform the following functions:

- Generate their own public-private key pairs
- Verify the quality of the public key parameters selected
- Create and deliver subordinate certificates
- Ensure there are no distinguished name collisions within local name space
- When issuing ORA certificates, subordinate CA certificates, and cross-certificates, verify that CAs or ORAs requesting the certificates are in possession of the private keys for all public keys submitted for certification.
- Sign and verify signatures
- Create, maintain, and distribute Certificate Revocation Lists (CRLs)
- Maintain record of certificates issued
- Create and maintain system audit logs
- Archive certificates and CRLs
- Generate or obtain time stamps
- Revoke certificates

CAs need to verify the identity of the originator of certificate requests prior to issuing certificates. Two forms of certification requests will be supported: initial, whereby the identity of the requestor is established in person at request time; and renewal, whereby the established identity of the requestor is verified by the digital signature on the request. Requests for new user certificates (i.e., not renewals) are always generated by an ORA function that vouches for the identity of the user. The ORA function is responsible for providing and verifying all the required personal and affiliation identification information for the type of certificate requested.

Upon receipt of an initial certificate request, CAs: (1)verify the signature of the ORA and that the information on the request is accurate, (2) complete the certificate and sign it with the CA's private key, (3) post the new certificate on a Directory, and (4) return the new certificate along with the CA's own certificate. The certificate may be either returned to the ORA, who then delivers it to the user, or directly to the user. Depending on the CA Operational Policy, the CA may actually return the certificate to both the ORA and the user, thus allowing the ORA to keep record of the certificates issued.

CAs are expected to operate in physically secure environments. The generation of CA private keys, and the hashing/signing of certificates and CRLs occur within cryptographic modules as defined in FIPS 140-1 [6]. In general, the assurance level for an ORA should not be allowed to limit that of the CA, this could be achieved either through security features of the ORA or physical protection and controls. CA and ORA agents are instructed on the operation of their respective systems and provided with reference material on the proper use and safeguard of key material, audit logs, personal information, and archival material. CA and ORA agents are also instructed on the rules and procedures for reporting lost or compromised keys.

Requirements for Low Assurance CAs

Low assurance CAs may only issue certificates that support low-risk applications, such as electronic mail.

These CAs may be implemented on systems conforming with FIPS 140-1 Level 2 security requirements and operated by a single CA agent. Keys used for signing certificates are never exported in clear form and should reside in a hardware token under the control of the CA agent.

Requirements for Medium Assurance CAs

CA cryptographic modules must minimally conform to the Level 2 requirements of FIPS 140-1. In addition, medium assurance CAs must provide direct key entry for the input of unprotected key components, separate ports (or pins) for entering plaintext authentication data or keys, and identity-based authentication (all Level 3 requirements). Private keys either remain stored within a cryptographic module or are enciphered using a FIPS-approved algorithm, and cryptographically split, before being output. Security practices such as separation of privilege must be employed.

Requirements for High Assurance CAs

Cryptographic modules for high assurance CAs are implemented in hardware and meet FIPS 140-1 Level 3 requirements.

Certificate Issuance Policies

Certificate issuance policies state the requirements or constraints under which certificates are issued. This includes (1) the personal identification requirements for regular users, subordinate CA agents, and ORA agents being certified, (2) procedures for the generation, safe keeping, revocation, and archiving of key material, and (3) an optional statement of the community for which a CA intends to issue certificates. The TSP specifies issuance policies for CA certificates , ORA agent certificates, and three types of user certificates. Low assurance level user certificates are called L-type certificates, medium assurance user certificates are called M-type certificates, and high assurance level user certificates are called H-type certificates. There are basic similarities between the issuance policies for all these certificate types, the main differences are in the rigor of the identification and authentication requirements, certificate validity periods, key sizes, and number of certificate renewals allowed. The issuance policy details not discussed here are determined by the specific policies of each CA.

For initial certification, CA and ORA agents and users identify themselves in person to the issuing CA. CA and ORA agents identify themselves by presenting their organization's picture id and a letter from a recognized sponsor identifying them as agents. Government users identify themselves by presenting their organization's picture id, other users present any Government issued picture id (e.g., drivers license, passport). Once requesters establish their identities with the issuing CA, or its ORA, they provide a self-signed skeleton certificate containing the public key (i.e., a certificate request). Certificate requests for CA, ORA agent, and H-type certificates must be presented to the ORA on hardware cryptographic tokens, those for other certificate types may be presented on a diskette. The hardware cryptographic tokens used by users requesting certificates for high assurance CAs and H-type user certificates must minimally conform to FIPS 140-1 Level 3. These cryptographic tokens must be unable to export the signature private key.

For every user or agent requesting a certificate of any type, except possibly for L-type user certificates, the CA must receive a request from a recognized user sponsor to issue that certificate. These requests are made through out of bands means and usually consist of a list of names with identification information and the type of certificate requested. ORAs instruct and/or train users, at a level appropriate to the assurance level of their certificates, on the proper use and safeguard of their PKI clients and key material, including rules for reporting lost or compromised keys.

Certificate renewal requires an electronic request signed with both the current, unrevoked, private key and the new signature key. The double signature binds the new key to the existing certificate and allows the parent CA to verify that the requester possesses a valid new key pair. CA policies state how many times certificates of each type may be renewed. Revoked certificates may not be renewed; replacement of revoked certificates must follow the initial certification procedure. The required signature key sizes and their validity periods for each type of certificate are also determined by CA policies.

Federal Assurance Levels

A Federal Assurance Level is an indication of the general level of trust that can be placed on a certificate that will be broadly understood throughout the Federal PKI. The assessment of the trustworthiness of the information in a certificate is made by the PAA upon evaluating the policies and procedures followed by the certifying CA. This effectively maps the actual CA Operational Policy and Issuance Policy followed in generating each certificate onto a Federal Assurance Level. Although Federal Assurance Levels will be conveyed in the certificate policy extension of Federal PKI certificates, they are not actual policies. The three Federal Assurance Levels defined in the TSP (low, medium, high) will be registered by the CSOR under the certificate policies branch. A single Federal Assurance Level will be assigned to every certificate.

Policy Approval Authority (PAA)

A Policy Approving Authority (PAA) is the policy approval and enforcement entity for a specific domain within the Federal PKI. It is responsible for the oversight of the operations of all infrastructure components in its domain. The PAA is directly associated with the root CA for its domain, but it delegates oversight responsibilities to subordinate authorities. The PAA evaluates CA Operational Policies and Certificate Issuance Policies to assess the overall quality of the certificates issued by each CA. This assessment is based on the guidelines outlined in the TSP [5]. The PAA conducts periodic reviews on a periodic basis that it establishes and may revoke the certificates of Federal CAs that fail to implement certificate generation and maintenance procedures in accordance with their own policies. The PAA authorizes Federal PKI CAs to include Federal Assurance Level identifiers in the certificates they issue based on that assessment.

Additional Policy Guidance

Records Keeping

Each CA will log the following certification activities: request to create a certificate, certificates issued, request to revoke a certificate, generation of a CRL, and distribution of a CRL to a Directory. Once a week this information will be stored off-line for archival purposes. All archived information will be maintained in a form that prevents unauthorized modification. Every CA must keep a separate audit log for the monitoring and tracking of security incidents.

Backups

As a minimum all CAs and ORAs within the Federal PKI should conduct daily system backups.

Notification Procedures

Upon occurrence of a system compromise or failure that may affect the integrity of the infrastructure, the CA affected must obtain new certificates, issue the appropriate CRLs, and notify the affected parties of the need to re-authenticate to replace the compromised certificates.

Initialization Procedures

Upon system startup each CA must obtain the certificates it needs from the parent CA and then issue certificates to all its users and subordinate CAs. As new certificates are generated, the Directory server is notified and populated with valid certificates. If the number of users is large, the process may take place in stages. When initialization occurs after a system compromise or failure, an effort will be made to issue notifications to the subscribers if delays are expected to extend beyond 24 hours. Steps must be taken to minimize the possibility of compromise and corruption at all levels of the PKI and to expedite recovery procedures.

Certificate Revocation

CAs will revoke a certificate after validating a request from the certificate holder (a user, CA, or ORA), the ORA that requested the certificate, or the PAA. Common reasons for requesting revocation are: change of the owner's name, separation from the issuing organization, change of the privileges of the user, or failure by a CA to demonstrate compliance with its policies or to implement appropriate operational procedures.

User revocation requests may be directed to either the CA or the ORA. The CAs will accept electronic revocation requests signed with the key being revoked, revocation requests presented in person, or through the telephone. All revocation requests must be verified by the CA prior to taking effect. When the request is made in person, the user needs to provide appropriate identification and the reason for the revocation. Revocation requests over the telephone can only be accepted if a satisfactory personal identification can be made. CAs will issue an electronic notification of the request to the user's superior or agency sponsor. Electronic revocation requests also require verification by the CA.

After processing a request for revocation, the CAs will update and sign the CRL. CAs transmit CRLs to a Directory twice daily, if any new revocations have occurred, or at least once every three days. CRLs are always signed by the issuing CA. Expired certificates are deleted from the CRL.

Certificates are also revoked as part of recovery from compromise or database corruption. If the CA suspects database corruption, in addition to the key compromise, it must revoke all subordinate certificates and electronically notify the subordinates. Old subordinate certificates need not be put on a CRL since the signatures on them will not verify. Subordinates can get the CA's new public key from the Directory. Replacement of revoked certificates is accomplished by following the procedure used for initial registration.

Cross Certification

Cross certification is a mechanism in which two CAs grant each other certificates to signify a trust relationship. This differs from the strict hierarchy model where trust is passed down hierarchically along single certificate paths. The Federal PKI is organized as a hierarchy for administration purposes, but allows the establishment of cross-certificates with some restrictions. The use of cross-certificates allows the establishment of a network of trust relationships among CAs within and outside the Federal hierarchy.

The Federal PKI defines three types of cross-certificates: hierarchical, general, and special. Hierarchical cross-certificates parallel the hierarchical path from the root CA. In the Federal PKI, every CA must trust its parent CA. At certification time every CA cross-certifies its parent to ensure the existence of at least one cross-certificate path to that CA from other Federal CAs. General cross-certificates are intended to simplify certificate paths for efficiency reasons (i.e., to shorten the paths) and may not allow the circumvention of restrictions. Special cross-certificates are intended to establish a relationship between two CAs, not allowed by hierarchical restrictions to certify subordinate CAs. These are called leaf CAs. Using special cross-certificates CAs may circumvent many of the restrictions imposed by their hierarchies. For instance, they could relax the name restrictions imposed by the hierarchical path, or grant each other cross-certificates with an assurance level higher than that of the certificates granted by their respective parent CAs. Only leaf CAs are allowed to establish special cross-certificates to ensure that the circumvention of hierarchically imposed controls is limited to the users of the CAs involved.

Conclusion

The proposed Federal Public Key Infrastructure needs to accommodate the use of dissimilar security policies while providing uniform levels of service and supporting on-line decisions to accept a digital signature. The policies for the Federal PKI deal with the generation, deactivation, and dissemination of public key certificates, the integrity of the infrastructure, maintenance of records, identification of certificate holders, and the establishment of trust relationships between Certification Authorities (CAs). Besides verifying a digital signature, the verifier needs to factor the trustworthiness of the CAs involved in the certification of the sender to determine the trustworthiness of an electronic message or data file. To accomplish this, the verifier needs to examine the certificate policy for those CAs. The Federal PKI Technical Security Policy establishes basic policies for the operation of Federal CAs and the identification of the parties requesting certification. It also creates a management entity that will police the operation of Federal CAs and assess the assurance levels for each CA. These assurance levels can be used in lieu of a certificate policy making an on-line determination of the trustworthiness of a certificate.

References

1. Burr, Nazario, Polk; *A Proposed Federal PKI using X.509 V3 Certificates*, Proceedings from the National Information Systems Security Conference, September 1996.

2. CCITT X.500 Series (1993) | ISO/IEC 9594,1--9, *Information Technology -- Open Systems Interconnection -- The Directory*, 1995.

3. DOD 5200.28-STD, *Department of Defense Trusted Computer System Evaluation Criteria*, December 1985.

4. Draft Amendments to ITU-T Rec. X.509 | ISO/IEC 9594-8, *Information Technology -- Open Systems Interconnection -- The Directory: Authentication Framework*, August 1995.

5. *Federal Public Key Infrastructure (PKI) Technical Specifications (Version 1) - Part B: Technical Security Policy*, Federal PKI Technical Working Group, March 13, 1996.

6. FIPS PUB 140-1, *Security Requirements for Cryptographic Modules*, January 1994.

7. NISTIR 5308; N. Nazario; *General Procedures for Registering Computer Security Objects*, December 1993.

A PROPOSED FEDERAL PKI USING X.509 V3 CERTIFICATES

William E. Burr, Noel A. Nazario and W. Timothy Polk
National Institute of Standards and Technology
Gaithersburg MD, 20899

1. Introduction

Public key certificates and digital signatures allow parties who were previously unknown to each other to establish trust relationships and possibly conduct secure, encrypted communications. The Federal Government is a large user community that could greatly benefit from this technology. A public key infrastructure (PKI) is needed to enable broad use of certificates across and among such large user communities.

Early attempts to establish public key infrastructures based on the X.509 public key certificate standard, such as Privacy Enhanced Mail (PEM) [RFC 1422] and the DoD Multi-level Information System Security Initiative (MISSI) [MISSI 95], have defined a hierarchical structure for the infrastructure. Although the hierarchical model is reasonably congruent with the structure of the Government and many other organizations, the primary advantage of the hierarchy was that it provided a convenient way to manage trust and security policies. That is, various branches of the tree have consistent security policies, and the level of trust assigned to a certificate holder can then depend upon the branch of the tree.

As standards for public key certificates evolve, a strict hierarchy is seen as unacceptably inflexible and hierarchical PKIs have not been widely implemented. The "version3" revision to the CCITT X.509 certificate standard [DAM95] extends the certificate with provisions that facilitate explicit management of certificates, certification paths, security policies, and the transfer of trust, so that non-hierarchical infrastructures are now practical and manageable.

This paper describes a proposed structure for a Federal PKI, developed by the Federal PKI Technical Working Group and stated in the Federal PKI Concept of Operations [CONOPS 95], that combines a hierarchy with a more general networked cross-certificate structure. It offers most of the advantages of both systems. A trusted entity that issues public key certificates is called a *certification authority (CA)*. An important attribute of this proposal is that a local CA may issue certificates and broadly cross-certify with whomever it needs, but the certificate holders of other CAs are protected from the possibly unwise cross-certification decisions of that CA.

2. Public Key Certificates

Figure 1 illustrates the X.509 v3 certificate. A certificate includes the issuer name, the subject name and the subject's public key, and is signed with the issuer's private key. If Alice has Bob's certificate, and knows the issuing CA's public key, she can verify Bob's certificate and then use Bob's public key to verify Bob's signature on any document.

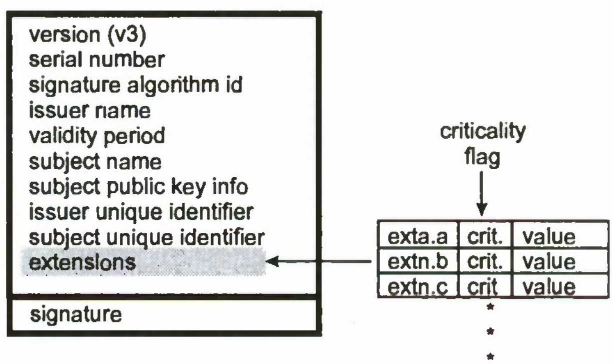


Figure 1 - X.509 Version3 Certificate

Table 1 - Standardized Certificate Extensions

Extension	Used By	Use	Critical (see Note)
Key and Policy Information			
authorityKeyIdentifier	all	identifies the CA key used to sign this certificate	No
keyIdentifier	all	unique with respect to authority.	
authorityCertIssuer	all	identifies issuing authority of CA's certificate; alternative to key identifier	
authorityCertSerialNumber	all	used with authorityCertIssuer	
subjectKeyIdentifier	all	identifies different keys for same subject	No
keyUsage	all	defines allowed purposes for use of key (e.g., digital signature, key agreement...)	Opt.
privateKeyUsagePeriod	all	for digital signature keys only. Signatures on documents that purport to be dated outside the period are invalid.	Opt.
certificatePolicies	all	policy identifiers and qualifiers that identify and qualify the policies that apply to the certificate	Opt.
policyIdentifiers	all	the OID of a policy.	
policyQualifiers	all	more information about the policy	
policyMappings	CA	indicates equivalent policies	No
Certificate Subject and Issuer Attributes			
subjectAltName	all	used to list alternative names (e.g., rfc822 name, X.400 address, IP address...)	Opt.
issuerAltName	all	used to list alternative names	Opt.
subjectDirectoryAttributes	all	lists any desired attributes (e.g, supported algorithms)	Opt.
Certification Path Constraints			
basicConstraints	all	constraints on subject's role & path lengths	Yes*
cA	all	distinguish CA from end-entity cert.	
pathLenConstraint	CA	number of CAs that may follow in cert. path; 0 indicates that CA may only issue end-entity certs.	
nameConstraints	CA	limits subsequent CA cert. Name space.	Opt.
permittedSubtrees		names outside indicated subtrees are disallowed	
excludedSubtrees		indicates disallowed subtrees	
policyConstraints	all	constrains certs. Issued by subsequent CAs	Opt.
policySet	all	those policies to which constraints apply	
requireExplicitPolicy	all	All certs. Following in the cert. Path must contain an acceptable policy identifier	
inhibitPolicyMapping	all	prevent policy mapping in following certs.	
CRL Identification			
crlDistributionPoints	all	mechanism to divide long CRL into shorter lists	Opt.
distributionPoint	all	location from which CRL can be obtained	
reasons	all	reasons for cert. inclusion in CRL	
cRLIssuer	all	name of component that issues CRL.	

NOTE: " "No" means the standard requires the extension be noncritical if used, and "Opt." means that the issuing CA may choose to make that extension either critical or noncritical. "Yes*" means that the standard allows the field to be either critical or noncritical, but the recommendation for the Federal PKI is that it be set to critical. There are no v3 certificate extensions that are required by the standard to be critical.

The optional extensions field is new in the v3 certificate. A certificate can hold any number of extensions. Each extension has a "criticality flag." If a certificate contains a critical extension, a certification path verifier that attempts to verify that certificate must be able to process that extension, or must not verify the certificate. A number of extensions are being standardized [DAM 96]. These standardized extensions are summarized in Table 1. In this paper sans serif type is used to identify the formal names of standardized extensions (e.g., policyConstraints).

3. PKI Organization

Certificates may be chained to form a *certification path*. This is illustrated in Figure 2; Bob has been issued a certificate by CA 3, which has been issued a certificate by CA 2, which in turn has been issued a certificate by CA 1. If Alice trusts CA 1and knows its public key, she can verify each certificate in the certification path until she reaches Bob's certificate and verifies it. At that point, Alice now knows Bob's public key and can verify his signatures.

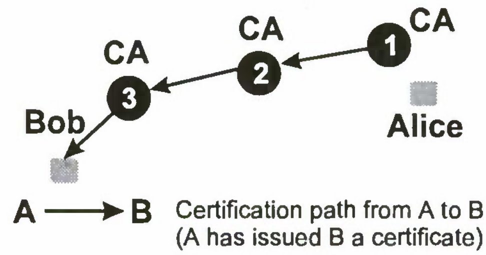


Figure 2 - Certification Path

CAs can certify each other in some systematic manner to form a PKI. A CA may be issued a certificate by another CA. Two CAs may issue each other certificates; this is known as *cross-certification*, and the pair together is a *cross-certificate*. Two alternative PKI topologies, illustrated in Figure 3 below are:

- *Hierarchical*: Authorities are arranged hierarchically under a "root" CA that issues certificates to subordinate CAs as illustrated in Figure 3 (a). These CAs may in turn issue certificates to subordinate CAs, or to users. Every user knows the public key of the root CA, and any user's certificate may be verified by verifying the *certification path* that leads back to the root CA. Alice verifies Bob's certificate, issued by CA 4, then CA 4's certificate, issued by CA 2, and then CA 2's certificate issued by CA 1, the root, whose public key she knows;

- *Network*: Independent CA's cross-certify each other, resulting in a general network of trust relationships between CAs. Figure 3 (b) illustrates a network PKI. A user knows the public key of a CA near himself, generally the local CA that issued his certificate, and verifies certificates by verifying a certification path that leads back to that trusted CA. For example, Alice knows the public key of CA 3. There are several certification paths that lead from Bob to Alice, but the shortest requires Alice to verify Bob's certificate, issued by CA 4, then CA 4's certificate issued by CA 5 and finally CA 5's certificate, issued by CA 3. CA 3 is Alice's CA and she trusts CA 3 and knows its public key.

The hierarchical PKI architecture has some advantages. The structure of many organizations such as the government is largely hierarchical and trust relationships are frequently aligned with organizational structure. A hierarchical PKI may be aligned with hierarchical directory names and the certification path search strategy is straightforward. Each user has a certification path back to the root; the user can provide this path to any other user and any user can verify the path, since all users know the root's public key.

It is likely, however, that the strongest reason why early PKIs have been hierarchical is that the hierarchy can be aligned with security policies and this alignment can be used to manage and determine the trust accorded to a particular certification path. While earlier versions of X.509

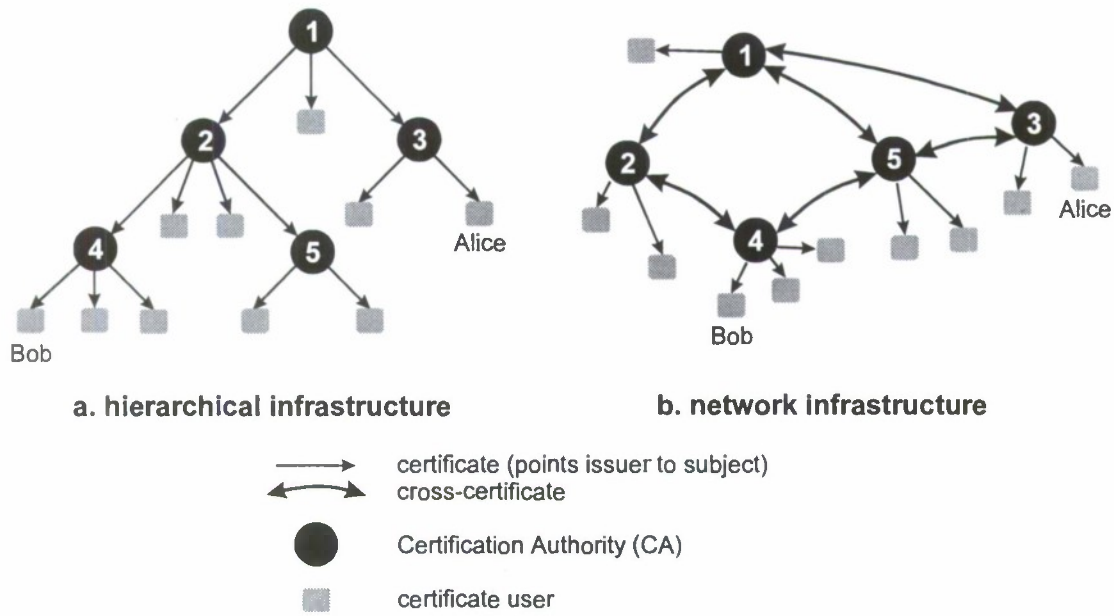


Figure 3 - Alternative PKI Topologies

allowed networks of cross-certified CAs, they provided no mechanism to manage trust in such networks. Version 3 certificates provide alternative means for managing policies and trust.

A strictly hierarchical certification path architecture has some disadvantages. It is improbable that there will be a single root CA for the world, therefore cross-certificates must exist at some level, and certification path verifiers must be able to cope with topologies that are not entirely hierarchical. Commercial and government trust relationships are not necessarily hierarchical, so using the hierarchy itself to manage trust relationships is surely not optimal. Moreover, compromise of the root private key is catastrophic because every certification path is compromised and recovery requires the secure "out-of-band" distribution of the new public key to every user;

The network certification path architecture has the advantage that it is flexible, facilitates ad hoc associations and trust relationships, and readily reflects bilateral trust relationships. It is likely that a national or worldwide PKI will evolve in an ad hoc fashion, from isolated CAs, and this is more easily accommodated in a network than a hierarchy. CAs that are organizationally remote, but whose users work together with a high degree of trust, can be directly cross-certified under a high trust policy that is higher than would be practical through a long, hierarchical chain of certificates. The CAs whose users communicate frequently, can cross-certify directly, reducing certification path processing.

Perhaps the most compelling argument for a network PKI is that it is more convenient and natural for a certificate holder to place his trust in the local CA that issued his certificate, rather than a remote root CA, and make this the foundation of all trust relationships. Moreover, this simplifies the out of band secure distribution of the CA public key and recovery from the compromise of any CA's private key now requires only that the new public key be securely distributed to the holders of certificates from that CA, and new certificates be generated for them.

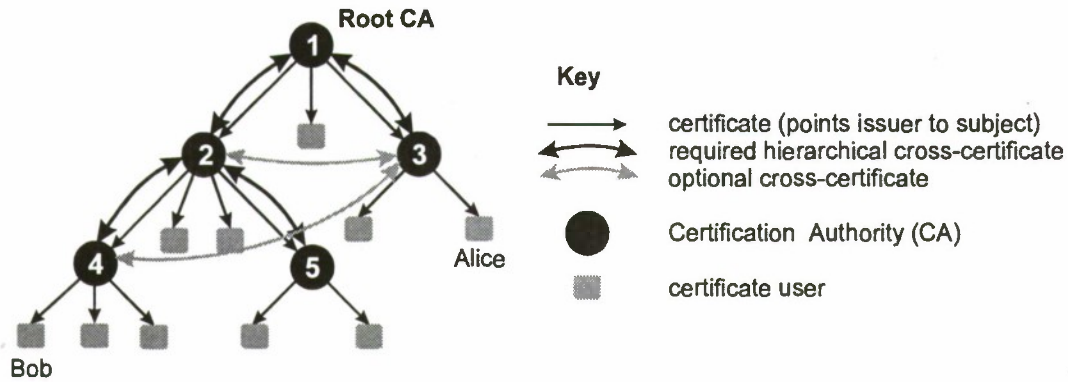


Figure 4 - Proposed Federal PKI Certification Path Architecture

The network PKI has at least two disadvantages: (1) Efficient certification path search strategies are more complex, and (2) a user cannot provide a single certification path that is guaranteed to enable verification of his signatures by all other users of the PKI.

4. Combined Hierarchical-Network Federal PKI

The hierarchical and network PKI architectures are not mutually exclusive. The following hybrid certification path architecture, illustrated in Figure 4, is proposed for the Federal PKI:

- There will be a hierarchical path of certificates leading from the root CA to its subordinate CAs, and from each of these CAs to their subordinates, and so on, until every Federal end user is issued a certificate with a certification path from the root CA;
- Each Federal CA will have a single parent. There will be one or more instance of the directory attribute certificate for certificates issued by the parent. There will be only one hierarchical path to the root CA based on the directory attribute certificate. Other certificates held by a CA, from any other issuer, will be posted in the directory in a crossCertificatePair;
- In parallel to the certificates hierarchically linking CAs to the root will be crossCertificatePairs attributes also linking those CAs. These parallel crossCertificatePairs are required and are shown in Figure 4 as black double-headed arrows. This will allow client applications that perform certification path verification from the verifier's parent CA, using the crossCertificatePair directory attribute, to operate from any Federal CA;
- Federal CAs may cross-certify each other along paths that do not parallel the hierarchy. Optional crossCertificatePairs are shown in Figure 4 as gray double-headed arrows.

If Alice now wishes to verify Bob's signature, she can find either a certification path that relies on her trust in her parent CA, CA3, or Bob's certification path back to the root. In general, Federal PKI clients and applications may choose to follow either a certification path verification strategy that leads to the root CA, or back to their own CA. Because of the hierarchical cross-certificates, a certification path is guaranteed to exist from her own CA, through the root CA, to every Federal certificate, but there may also be much shorter paths.

5. Federal PKI Management

Some overall management of Federal CAs is needed if trust is to be broadly propagated in an organization as large and diverse as the Federal Government. In this proposal overall management of the Federal PKI is assigned to a Policy Approving Authority (PAA) associated with the root CA. The proposed management principle is to exercise only the central control needed to ensure broad, consistent transfer of trust throughout the Federal PKI and to limit the damage that holders of certificates from one Federal CA are exposed to as a result of the actions of another CA, while still allowing all Federal CAs broad discretion to serve their users as they see fit.

5.1 Use of V3 Extensions

This proposal uses three extensions to implement government wide management in the Federal PKI:

- certificatePolicies: certification path verifiers compare a list of acceptable policies to the policies listed in the certificate. If there is no match, verification fails. Use of this extension is described in section 5.2 below;
- nameConstraints: this critical extension constrains a CA to issue certificates only for the namespace of specified directory subtrees. Several subtrees can be included. The PAA may use the nameConstraints to restrict namespace for which CAs immediately subordinate to the root may issue certificates, and they may further restrict their subordinates;
- pathLengthConstraint: this component of the critical basicConstraints extension limits the number of certificates that may follow in a certification path. A CA whose certificate pathLengthConstraints value is zero may issue only end entity certificates. The PAA may assign a pathLengthConstraint to certificates issued by the root CA, to limit certification path lengths. Special requirements for cross-certificates are stated in section 5.3, below.

5.2 Policies

We propose that every CA in the Federal will have a PAA approved *operational policy*, governing how the CA is operated (e.g., how the CA private key is protected, how the CA is physically protected, how data is backed up, etc.), and one or more PAA approved *certificate issuance policies*, governing how the CA issues certificates. A principal features of a certificate issuance policy is how the identity of certificate subjects is verified.

V3 certificates allow a policy identifier to be placed in the certificatePolicies extension. If there are many different policies, automatic verification will not be practical. A small set of policy identifiers called *Federal-Assurance-Level-IDs* will be defined (initially, *high, medium* and *low*) for Federal use to indicate a relative assurance level, and one of these will be included in the certificatePolicies extension of every FEDERAL PKI certificate. The PAA will evaluate each CA operational policy and certificate issuance policy pair, and determine the highest Federal-Assurance-Level-ID that may be assigned to certificates issued under that policy pair.

5.3 Cross-Certificate Management

Cross-certificates are contained in the directory attribute crossCertificatePair. When CA X cross certifies with CA Y, the directory entry for CA X holds a crossCertificatePair containing two certificates, one called forward, containing the certificate issued by X to Y, and one labeled reverse, containing the certificate issued by Y to X. In Y's directory entry there is a "mirror image" crossCertificatePair.

The essential issue with cross-certificates is how to allow CAs to cross-certify with other CAs to meet the particular needs of their own users, without compromising the security of users of other CAs in the Federal PKI. For example, a particular agency might have a close working relationship with a local government office, a particular contractor, or law firm that has its own CA. That relationship, however, would not necessarily justify extension of trust to other government agencies. To accomplish this three classes of cross-certificates are proposed below for the Federal PKI.

5.3.1 Hierarchical cross-certificates

Hierarchical cross-certificates exactly parallel the hierarchical certification path to the root CA. The forward certificate of each crossCertificatePair for a parent CA is the certificate it issues to the subordinate CA. These hierarchical cross-certificates, shown in Figure 4, are used to ensure that clients that verify certification paths from their own CA, can always find a certification path to any certificate issued in the Federal CA.

5.3.2 General cross-certificates

General cross-certificates supplement the certification hierarchy and allow shorter certification paths. General cross-certificates are governed by rules, described below, so that, when they are used, the propagation of trust is equivalent to the trust that would result from the use of the hierarchical certification paths to the root CA. They are appropriate when cross-certification will shorten the certification paths and improve performance of frequently used paths. In Figure 4, the cross-certificate between CA 2 and CA 3 is a general cross-certificate.

The rule for certificates issued by Federal CAs as part of general cross-certificates is that, before issuing the certificate, the issuer first evaluates the hierarchical certification path from the subject CA to the root CA. It then includes values for certifictePolicies, pathLengthConstraint and subtreesConstraint as follows:

- certificatePolicies: the value of the Federal-Assurance-Level-ID included in a certificate issued as a part of a general cross-certificate is not greater than the lowest Federal assurance level found in the path back to the root.
- pathLengthConstraint: the value contained in a certificate issued as a part of a general cross-certificate is not greater than the path length remaining on the path from the root.
- subtreesConstraint: the values contained in a certificate issued as a part of a general cross-certificate are at least as restrictive as the constraints inherited by the CA along the path from the root. General cross-certification between Federal and non-Federal CAs requires that the certification path to the root CA allow issuance of certificates to non-Federal names.

The effect is that any certification path that includes a general cross-certificate has path length and subtrees constraints at least as restrictive as those imposed through the hierarchical path from the root, and the highest Federal Assurance Level supported by a path using a general cross-certificate is not greater than the highest level supported by the hierarchical path from the root.

5.3.3 Special cross-certificates

Special cross-certificates allow certification paths that do not conform to the restrictions imposed hierarchically along the path from the root CA. Special cross-certificates may only be created between "leaf" CAs, that is CAs with a zero pathLengthConstraint value in all certificates issued to it by other Federal CAs. This blocks further propagation of trust to another CA along the hierarchical certification path. In Figure 4, the cross-certificate between CA 3 and CA 4, both leaf CAs, is a special cross-certificate. A pathLengthConstraint value of zero is included in the two certificates of special cross-certificates to prevent concatenation of special cross-certificates.

Because of the pathLengthConstraint in all the leaf CA's certificates, only the users of certificates issued by the two CAs participating in the special cross-certificate may use the less restrictive certification path. With special cross-certificates, users of the two CAs may operate under policies allowing a higher trust level or less restrictions than would otherwise be permitted. For example, a CA X, holding a certificate from its parent with a subtreesConstraint that limited its name space to the Department of Commerce, could cross-certify with a non-government CA. Holders of certificates issued by other government CAs could not use that special cross certificate in Certification paths for two reasons: (1) it violates the subtreesConstraint of CA X's own certificate, and (2) the pathLengthConstraint of CA X's own certificate prevents use of the cross-certificate. Holders of certificates from CA X, who verify certification paths through CA X's public key, would not encounter these constraints.

6. Conclusion

Prior to the advent of v3 certificates, attempts to design large public key infrastructures had featured a hierarchical organization of CAs and certification paths. The main reason for this was to facilitate the management of trust relationships by aligning them with the hierarchy. Certification path verifiers in a hierarchical infrastructure rely on the public key of the root CA. This, however, is an inflexible architecture for large, diverse organizations such as the US Federal Government, and it is difficult imagine how to connect together independent CAs around the world hierarchically. Who would operate the root CA?

The latest revision of the X.509 certificate standard includes several extensions that can be used to manage trust relationships in an architecture of cross-certified CAs, which use client certification path verifiers that rely on the public key of the CA that issued the client his certificate. This is more flexible, and facilitates the growth of an ad-hoc national or international PKI of cross-certified CAs, as needed by individual CAs. It does not, however, automatically provide a framework for coherent overall management of trust relationships in a large organization such as the US Federal Government.

This paper describes a hybrid certification path architecture, developed by the Federal PKI Technical Working Group, that preserves many of the advantages of each architecture, and is proposed for use in a Federal PKI. This architecture uses a hierarchical structure with the new certificate extensions to allow overall management of trust relationships, while giving individual agency CAs the flexibility to cross certify with other Federal and non-Federal CAs as needed to meet the needs of their users. In particular, it prevents unwise cross-certifications of one Federal CA from compromising users of other Federal CAs. It also supports the use of certification path verifiers and trust models that rely on the public key of either the root CA, or the local CA.

7. References

[CONOPS 95] *Public Key Infrastructure (PKI) Version 1 Technical Specifications - Part C: Concept of Operations*, Federal PKI Technical Working Group, Nov. 16, 1995.

[DAM 96] ISO/IEC JTC 1/SC 21 document, *Draft Amendments DAM 4 to ISO/IEC 9594-2, DAM 2 to ISO/IEC 9594-6, DAM 1 to ISO/IEC 9594-7, and DAM 1 to ISO/IEC 9594-8 on Certificate Extensions, May 20, 1996.*

[MISSI 95] *MISSI Phase 1 Program Overview* , VERSION: 3.3, 17 Oct. 1995.

[POL 95] TWG-95-81, *Technical Security Policy for the Federal PKI*, 25 Aug. 1995.

[RFC 1422] S. Kent, *Privacy Enhancement for Internet Electronic Mail, Part II: Certificate-Based key Management*, IETF RFC 1422, Feb. 1993.

[X.500 93] CCITT Recommendation X.500, *The Directory*, 1993.

[X.509 93] CCITT Recommendation 509, *The Directory: Authentication Framework*, 1993.

A Proposed Federal PKI Using X.509 V3 Certificates

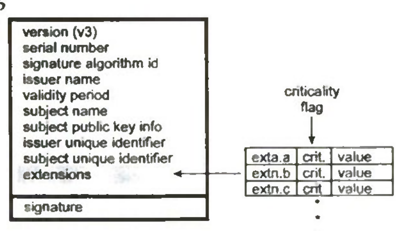
William E. Burr
Noel A. Nazario
W. Timothy Polk

NIST

1

X.509 v3 Certificate

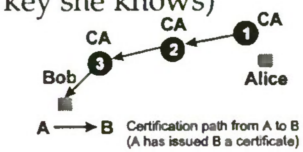
- Extensions
  - criticality flag
  - standardized extensions



2

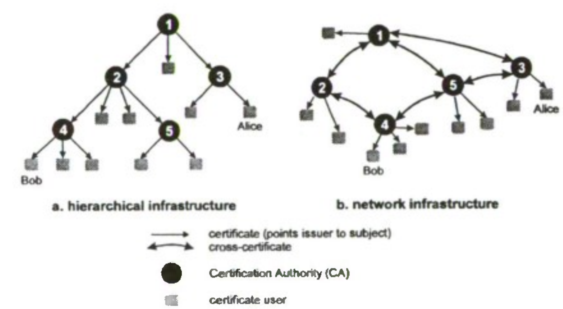
Certification Path

- Alice can verify Bob's certificate by verifying a chain of certificates ending in one issued by a Certification Authority (CA) she trusts (and whose public key she knows)



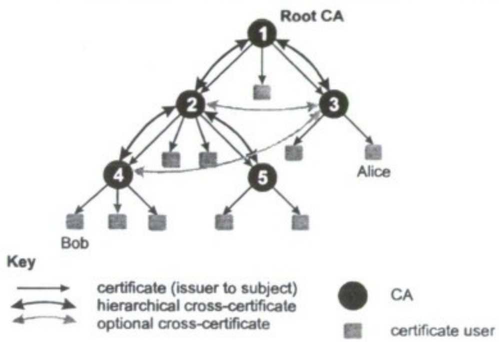
3

PKI: Hierarchy vs. Network



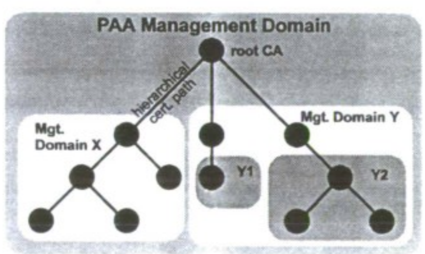
4

Hybrid Architecture



5

FPKI Management Domains



6

Hierarchical Management

- pathLenConstraint
  - limits length of certification path
- nameConstraint
  - limits name space a CA may issue certs. for
- certificatePolicies
  - Federal Assurance level
    - a system to give a relative level of trust

7

Federal Assurance Level

- OID goes in certificatePolicies extension in all Federal certificates
- Based on PAA evaluation of CA operational policy and certificate issuance policy
- States a relative trust assurance level
  - a few levels defined, such as: high, medium and low

8

Cross-Certificates: 3 Types

- hierarchical
  - parallels hierarchical certificates
  - uses superior to subordinate cert.
  - required
- general
- special

9

General Cross-Certificates

- May provide shorter cert. paths
- Allowed between any two Federal CAs
- Includes constraints at least as restrictive as those along root CA path
  - pathLenConstraint
  - nameConstraint
  - certificatePolicies
    - Federal Assurance Level (trust level indication)

10

Special Cross-Certificates

- Cannot be chained to other CA certificates to extend trust
  - Only between "leaf" CAs with "root certificate" pathLenConstraint of zero
  - Special cross-certificates have pathLenConstraint value set to zero
- Any other constraints are agreed between cross-certifying CAs

11

Conclusion

- Hybrid architecture
  - allows coherent management of FPKI
  - supports root or local CA centered trust models
- Special cross-certificates
  - allows CAs broad freedom to cross-certify
  - trust does not propagate to users of other CAs

12

A Security Flaw in the X.509 Standard

Santosh Chokhani
CygnaCom Solutions, Inc.

Abstract

The CCITT X.509 standard for public key certificates is used to for public key management, including distributing them with a high degree of confidence in binding between the users and their public keys. The two locations where the public key parameters of certificate signer (also called certificate issuer or certification authority) can be placed in a X.509 certificate are vulnerable to parameter substitution attack. The Department of Defense FORTEZZA card and the Multilevel Information Systems Security Infrastructure (MISSI) are **NOT** vulnerable to the attack described in this paper.

1.0 Introduction

The CCITT and ISO have developed a X.509 public key certificate standard to provide high integrity, authenticated binding between entities and their public keys. This standard is being adopted worldwide including the United States Federal Government, Government of Canada, American National Standards Institute (ANSI), and the U.S. banking industry for public key management and public key infrastructures. While there may be some minor differences in these standards, the security area analyzed in this paper is common to all of them. Hence, the findings of this paper are applicable to all known standards and implementations of public key certificates.

In Section 2, we provide a background on the X.509 certificate and certificate revocation list (CRL) standards. In Section 3, we describe the potential flaw the standard is vulnerable to. In Section 4, we describe the risk of the flaw based on various cryptosystems used to sign the certificates and CRLs. In Section 5, we provide some recommendations. Finally, an appendix provides some implications for the Digital Signature Standard (DSS).

2.0 X.509 Background

The joint ISO CCITT X.509 standard and its amendments describe the formats for public key certificate and CRLs issued by trusted authorities [4, 5]. These trusted authorities are also called Certification Authority or CA. The certificate and CRL are Abstract Syntax Notation.1 (ASN.1) encoded objects using the Distinguished Encoding Rules (DER). The entire content of the certificate and the ASN.1, DER concepts are not critical to understanding the flaw we describe. Thus, we will concentrate only on the aspect of the certificates and CRL that relate to the flaw. Figure 1 below describes the format of the

X.509 certificate. For the details of the contents of the certificate, please read the X.509 standards and related draft and balloted amendments. A public key certificate is a signed (by a CA) object that binds an entity (e.g., an user) to his/her public key. The certificate contents relevant to this paper are: certificate issuer (signer) distinguished name, subject distinguished name, and subject public key. This information is within the signed envelop of the certificate. The signed envelop may optionally contain issuer public key parameters and/or the subject public key parameters. In addition, as Figure 1 illustrates, the signature (termed signed macro in the X.509 standard) may optionally contain the issuer public key parameters. The signed macro always contains the digital signature. The inclusion of public key parameters in the signed macro allows efficient signature verification based on these parameters without having to decode the certificate and then extract the parameters from the issuer public key parameters field. The issuer public key parameters are included in the signed envelop and/or the signed macro to allow the CAs in a trust chain to have different public key parameters. The subject public key parameters field allows the subjects to have different parameters from their certificate issuers.

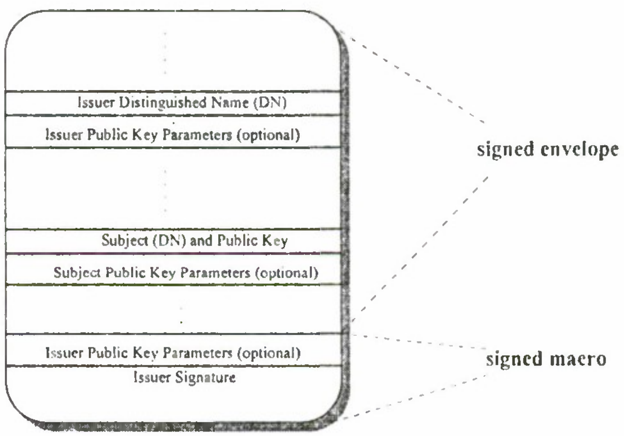


Figure 1: X.509 Public Key Certificate Format

Figure 2 below describes the format of the CRL. For the details of the contents of the CRL, please read the X.509 standards and related draft and balloted amendments. A CRL is a signed (by a CA) object that lists the revoked certificates. In order to maintain trust, public keys corresponding to revoked certificates should not be used since the CA no longer vouches for the binding between the users and their public keys as published in original certificates. The CRL content relevant to this paper is: certificate issuer (signer) distinguished name. This information is within the signed envelop of the CRL. The signed envelop may optionally contain issuer public key parameters. In addition, as Figure 2 illustrates, the signature (termed signed macro in the X.509 standard) may optionally contain the issuer public key parameters. The signed macro always contains the digital signature. The inclusion of public key parameters in the signed macro allows efficient signature verification based on these parameters without having to decode the CRL and then extract the parameters from the issuer public key parameters field. The issuer public key parameters are included in the signed envelop and/or the signed macro to allow the CAs in a trust chain to have different public key parameters.

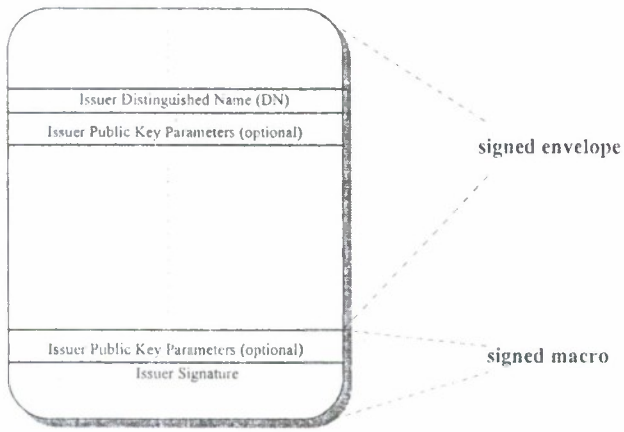


Figure 2: X.509 Certificate Revocation List Format

3.0 Basic Flaw -- Public Key Parameters Substitution

The use of issuer public key parameters fields (both in the signed envelop and in the signed macro) are vulnerable to substitution attack. The detailed scenario is as follows.

We need issuer public key and public key parameters to verify the signatures on the certificate and CRL. The issuer public key is expected to be obtained through a trusted and authenticated means. It is not available in the signed object (certificate and CRL).

A public key digital signature cryptosystem offer a certain degree of security. The degree of security is defined as the computational complexity of forging signatures or computing the private key for a public key and public key parameters of certain quality and size. For example, we know that in the Digital Signature Standard, the size of the large modulus p, size of the small modulus q, and the properties of p, p-1, and q are critical to security. The properties include ensuring that p and q are primes of appropriate size and that q divided evenly into p-1.

If the issuer public key parameters are used from the signed envelop or the signed macro, an attacker who wants to replace, modify or create bogus certificates and CRL, can substitute these values in the objects (certificate and CRL) and resign the objects (certificate and CRL). This allows the attacker to translate a hard public key cryptography problem into one of finding a new set of parameters and private key that are consistent with the trusted public key. Finding this may be easier, as hard or harder. This all depends on the mathematical properties of the cryptosystem.

For example in the DSS, the public key is y, private key is x, and public key parameters are p (large modulus), q (small modulus), and g (generator). We know that if the parameters are generated according to the standard, given y, p, q, g, it is hard discrete logarithm problem to find the private key x. What has not been analyzed in the literature is given y, could one find parameters p', q', and g' such that find a new key x' would be easier than the hard discrete logarithm problem. If this was possible, an attacker could substitute p, q, g in the issuer public key parameters in a certificate and/or CRL with p', q, g' and then use x' to sign the certificate and or CRL. The user of the certificate will use y, p', q', and g' to verify the signature.

In summary, our basic claim is that the two locations where the issuer public key parameters appear, are unauthenticated. This is true even if one of these paramcter set is within the signed envelop. This is due to that fact that the parameters values in the certificate itself are used to validate the signatures on the same certificate. Thus, an attacker can always substitute the parameters and resign. The ease of finding a private key and parameter set consistent with the authenticated public key depend on the cryptosystem chosen. The cryptosystem specific issues are analyzed in Section 4 below.

Impact of the Flaw

The flaw is extremely severe. It can destroy trust in an entire Public Key Infrastructure (PKI) since the attacker can modify or create bogus certificates and CRL for intermediate CAs in a chain and for end entities. The trust in a PKI and in a CA depends on the authenticity of certificates and CRLs.

4.0 Implications for Various Cryptosystems

RSA

The parameter substitution attack can not be used in X.509 certificates with RSA since the two public values required for RSA (e - encryption exponent, n - composite number) are both part of the public key. RSA has no public key parameters.

DSS

While the DSS is very clear on the requirement for the public parameters (p - large prime modulus, q - small prime modulus, g - generator) to be authenticated [1], some organizations have registered the DSS algorithms with ISO that provide for p, q, g to be parameters in X.509 sense. Thus, these parameters can be included in the two issuer parameters field discussed previously. Based on the analysis in Section 3 above, these values will be naturally unauthenticated. This leads to X.509 DSS based certificate implementations that are inconsistent with and are in contradiction with the specific requirement of the DSS, namely the need to use authenticated parameters. Appendix provides further details on how an attacker can substitute p, q, and g. The detailed mathematical analysis is beyond the scope of this paper.

MISSI

The attack described here can not materialize in the Department of Defense FORTEZZA card and MISSI due to the fact that MISSI always uses authenticated public key parameters and due to the cryptographic checks in the FORTEZZA card. MISSI uses the authenticated parameters for an initial trusted authority public key and only uses the parameters from the subject public key parameters in the certificates which are always authenticated due to the digital signatures on the certificate.

Different Meanings of the term "Public Key Parameters"

The term public key parameters in a cryptosystem generally means that they could be public and could be common to a group of users. For example, the term DSS parameters in the DSS standard are meant to convey elements of keying material that can be public and be common to a group of users. The DSS standard still requires these parameters to be provided in an authenticated manner and the cryptosystem security depends on their quality, size, and the users obtaining them in an authenticated manner.

The implication of the term "parameters" in the X.509 standard is bigger than the one in the DSS standard or potentially other cryptosystems. The implication in the X.509 standard is that the substitution of the parameter values (in issuer public key parameters fields) may not reduce the security of the cryptosystem. If the parameters are used in these fields, the security of the base cryptosystem can be changed to that of computing a private key that maps to the registered public key under the substituted parameters.

5.0 Recommendations

Analysis Based Parameter Definition

The X.509 certificates provide a flexible mechanism for registering public key and public key parameter syntax for various cryptosystems. When interested parties register a cryptosystem, the parameter substitution problem must be fully analyzed. If it can be shown that the substitution problem is at least as hard as the base cryptosystem, only then the parameters should be registered as part of public key parameters. If the analysis shows that the problem may be simplified or the answer is unknown, the parameters must be registered with the public key. The public key syntax must provide for optional inclusion of the parameters, in order to keep the certificate and CRL size small.

Ignore the Issuer Public Key Parameters Field in Registered Cryptosystem

For cryptosystems like DSS, where the parameters have been already registered and a preliminary analysis shows that the substitution attack is simpler than computing discrete logarithms for cryptosystems as defined in DSS, the parameters in issuer public key parameters fields must be ignored.

Change Cryptosystem Registry

For cryptosystems like DSS, where the parameters have been already registered and a preliminary analysis shows that the substitution attack is simpler than computing discrete logarithms for cryptosystem as defined in DSS, the registry should be modified to carry no parameters in the parameters field, but to carry them optionally in the subject public key information field only.

Use Parameters in Subject Public Key Parameters Field

Our previous recommendations do not reduce the flexibility of different users having different parameters. In a chain of certificates and CRL of arbitrary length, as long as one starts with authenticated public key and public key parameters of a trusted CA, and uses the values in the subject public key parameters field, the substitution attack will not materialize.

Check the Quality and Size of Parameters

One option is that during the use of a certificate or CRL (i.e., their verification) crypto engine checks the quality and size of unauthenticated parameters. We don't recommend this due its performance impact and since these checks may not be a sufficient substitute for authenticated parameters. For example, it will be take prohibitively long (at least minutes on a desktop workstation) to verify the primality of p and q in DSS.

Cross-fertilize

We stumbled into this flaw while developing rules for public key parameters inheritance in a certificate chain. One lesson we have learned is that the implementors need to pay greater attention to the security and mathematics of cryptosystems and the mathematicians need to be exposed to how the systems are being implemented. Otherwise, problems like this may go undetected.

References

1. FIPS PUB 186, May 19, 1994, page 7, Section 6.

2. Responses to NIST DSS Proposal, Ron Rivest, Communication of the ACM, July 1992, Page 43.

3. A Course in Number Theory and Cryptography, Neal Koblitz, Springer-Verlag, Second Edition.

4. Recommendation X.509 and ISO 9594-8, Information Processing System - Open Systems Interconnection - The Directory - Authentication Framework, 1988.

5. Final Text of Draft Amendment DAM 4 to ISO/IEC 9594-2, DAM 2 to ISO/IEC 9594-6, DAM 1 to ISO/IEC 9594-7, and DAM 1 to ISO/IEC 9594-8 on Certificate Extensions, April 1996.

Appendix - DSS Analysis

In this appendix, we offer some observations on the properties of the DSS in light of the X.509 flaw. A comprehensive mathematical analysis of the DSS cryptosystem is beyond the scope of this paper.

Some of the security aspects of p, q, and g in DSS are:

1. p to be a prime of appropriate size (i.e., $2^{511+64j} < p < 2^{512+64j}$) where j = 0,1,2,..,8.
2. q to be a prime of appropriate size (i.e., $2^{159} < q < 2^{160}$)
3. q evenly divides in p-1
4. g to be a power of (p-1)/q

It is anticipated that the digital signature verification software will not check any of the parameter properties. The primality tests for p, q are definitely out of question due to the time it takes to perform these checks. The security properties will be tested, if at all, during the key generation process. Furthermore, review of the standard shows that in order to generate valid signatures (i.e., the ones that can be verified) one only needs to ensure that p is prime and the property 4 above holds. Property 4 is trivial to meet if q need not be prime. It can be achieved by setting q = p-1 and making all generator satisfying the property since (p-1)/q = 1 and every integer's power of 1 is the integer itself. The rest of the requirements are not critical to mathematics of DSS; they are critical to the security of DSS.

A Simple Attack

The following is a simple attack. An attacker takes a trusted public key y and computes a new large prime modulus p > y. This is easy to do. The attacker sets q = p-1, h = g = y, and x = 1. Now, the attacker can masquerade as the public key "y" holder. This simple attack will change the digital signature components r, s from 160 bits each to the size of q (which is p-1) each.

Other Considerations

While one could develop simple parameters and public key test to prevent the above attack, there are other values the attacker can choose to simplify the discrete logarithm problem.

The following factors help an attacker create a realistic parameters substitution attack:

- weak and trap door prime p [2]
- q not being prime
- p-1 having all small prime factors, simplifying the discrete logarithm problem
- reducing the size of p to that of y, thus reducing the discrete log problem for smaller p
- x need not be constrained since only the attacker keeps x (private key).

According to [2], the DSS crypto problem is a variation of the classic discrete logarithm problem. We lack operational experience with ease of defeating the security of DSS.

The odds of getting a generator by random guess depend heavily on the factorization of p-1 [see page 35 in 3]. The probability that a random number is a generator is $\prod (1-1/l)$ over all l, where l's are the prime factors of p-1. Computing discrete logs is easy if all the primes dividing p-1 are small [see page 103 in 3]. That is one of the reasons for q to be a prime in DSS, guaranteeing that at least one of the prime factors of p-1 is large (160 bits in case of DSS). Since an attacker is generating new p, he may be able to control the probability of guessing a generator and simplifying the discrete logarithm problem. But, these two requirement (namely the ability to find a generator and the ability to compute discrete logarithms) seem to work against each other since too many small primes will make probability product defined above (for a random number to be a generator) small.

Acknowledgments

We found this flaw while developing algorithms for parameter inheritance in a certificate chain for the National Security Agency under contract. We appreciate the review of the drafts of this material by the National Security Agency, and Miles Smid and Jim Nechvatal of the National Institute of Standards and Technology. Miles Smid provided several insights for the organization of the paper and for the presentation of the findings.

COMPUTER VIRUS RESPONSE USING AUTONOMOUS AGENT TECHNOLOGY

Christine M. Trently

Mitretek Systems
7525 Colshire Drive
McLean, VA 22102

voice: (703)-610-1677 fax: (703)-610-1699
email: *ctrently@mitretek.org*

Abstract

Automating the computer virus response offers the ability to prevent and recover from computer virus incidents with minimal input from and impact on the user. This paper proposes an automated computer virus response capability using autonomous agent technology. Although autonomous agent technology has not been exploited in the anti-virus industry, its use in virus response can permit computer system environments to mimic the biological immune system by identifying viruses, removing viruses, and reporting virus incidents. This paper describes the potential use of autonomous agent technology for automating computer virus response, describes the functionality to be realized through the automated response, and then discusses the issues to be addressed for any automated system for handling computer virus response in an enterprise environment. Future directions and considerations for this research are also included.

KEYWORDS: Autonomous Agent; Computer Virus; Automated Response; Immune System

Introduction

During the past decade, the computer virus problem has reached worldwide recognition and prevalence. The 1995 Datapro Information Services Survey of Computer Security Issues showed that 32% of the respondents were extremely concerned with computer viruses and malicious code [2]. There are thousands of DOS viruses and the number is growing at an average of 3 new viruses per day [16]. However, only about 10% of the existing DOS viruses [8] have been seen in actual computer virus incidents or "in the wild" (ITW).

When reviewing the vast amounts of information available on the nature of computer viruses and the various anti-virus software products available, it became evident that computer viruses will be not going away in the near future [4]. In the 1996 Computer Virus Prevalence Survey compiled by the National Computer Security Association (NCSA), the number of virus exposures rose approximately ten-fold in the last year from one virus exposure for every thousand personal computers (PCs) per month to ten virus exposures for every thousand PCs per month [10]. The current mechanisms for detecting and recovering from the growing number of computer viruses are time consuming and require extensive awareness and training for the user community. It is no longer practical, particularly as the connectivity and interoperability advancements increase, to expect the average user to be extensively computer literate.

One manner in which to view the computer virus problem is to continue the comparison to its biological counterpart. The generation of an immune system for computers [7] can be further expanded to include the duplication of the biological equivalent of white blood cells or antibodies to combat "infections" as the computer or network is exposed to known virus strains. The antibodies in the biological immune system combat those entities that are foreign to the system, and the antibodies are not dependent upon one central source for knowing what to combat and how. This gives the antibodies the ability to be distributed and active throughout the body. Without the ability to be distributed and autonomous, the antibodies would be highly susceptible to attack because one entity that could disable one antibody would be able to disable any or all of them [3]. With the use of autonomous agents, the biological function of antibodies or an immune system can be realized in the automated environment.

Needing to More Fully Automate the Computer Virus Response

Since there are approximately 7000 viruses in existence worldwide [16], fully automating the computer virus response to such a large number of viruses is unrealistic and unnecessary. As noted above, only about 10% of the viruses in existence have actually be reported "in the wild." These are the viruses that can and should be handled in an automated fashion [8].

When looking at the effects of computer virus infections on an organization or enterprise, it is important to note that the costs associated with computer virus infections are growing as connectivity and interoperability increase and computer usage becomes more prevalent. These costs, which can be quite extensive in certain circumstances [10, 12], include the training of computer users in computer virus awareness and anti-virus product usage, the support of technical experts during a computer virus incident, and the interruption to productivity during an incident. In a 12 month period, 63% of the interruptions to processing in the microcomputer environment were attributed to computer viruses and malicious code [2].

The computer virus response within an enterprise includes:

- detecting and identifying the virus,
- collecting a sample of the virus (when possible),
- removing the virus,
- reporting the incident to an administrator or technical support, and
- keeping incident statistics.

These functions are currently performed by the user and require the user to be trained in the use of anti-virus products. Fully automating the response for ITW viruses [8] would seem to provide a considerable cost saving by eliminating the need for extensive training for the user and by reducing or eliminating the user productivity interruptions. An automated virus response could perform the detection, removal and reporting functions without interrupting or alarming the user [8]. Instead of notifying the user, an administrator is notified and the administrator can determine the extent of the incident as well as the need to inform the user. Automating the response, however, should not and does not abolish the need for general computer virus awareness information to be provided to any person using a computer.

A fully automated response, however, cannot be used in all computer virus incidents. The automated response, should, at least, detect and report all viruses, whether ITW, known or unknown. For those incidents dealing with previously unknown viruses, expert technical assistance will still be necessary.

Describing Autonomous Agent Technology

The term "agent" has been used and defined in a variety of ways. One such definition describes agents as "good viruses" [13] since the agent program acts in the background on behalf of the user and, in some instances, has the ability to replicate. Agents have also been compared to artificial life [9]. For this paper, however, autonomous agents are defined as a group of computer programs which utilize artificial intelligence techniques to fulfill a set of goals or tasks in a complex, dynamic environment [1]. Autonomous agent technology uses software designed to adapt its behavior based upon experience and from interactions with other agents in the environment. Each agent is designed to perform a simple, singular task. The collection of agents within an environment, however, can perform sophisticated, intelligent actions. In addition, the collection of agents can migrate throughout the computing environment performing tasks without any interference from or interaction with the user. The computing environment may be a single workstation or an entire network.

Agent Operating Environment

The operating environment for the autonomous agents needs to provide a mechanism for communication between the agents [5]. The agent operating environment can use the application programming interface (API) to pass information or parameters between the agents. In addition, the components of the agent operating environment need to be bound to various operating system functions [5]. These functions include such things as memory management, file management, and internal timing. The components of the agent operating environment also need to be bound to the available message transport service via the communications infrastructure to deploy and receive autonomous agents and their results. Once the components of the agent operating environment are established and bound to the communications infrastructure, the agents can perform their duties independently but have the results of their activities coordinated and managed.

Agent Coordination Engine

Since autonomous agents perform small, individual tasks, there is a need to coordinate the efforts performed and the results obtained by the agents [5, 6]. A centralized coordination engine running in the agent operating environment can provide the ability to coordinate and manage the flow and use of autonomous agents within a given system. The basic functions of an agent coordination engine (ACE) are depicted in Figure 1. The engine includes the ability to launch, authenticate, repair, and communicate with agents throughout the system. The functions of the ACE provide the autonomous agents with the ability to migrate throughout the computing environment to perform their tasks and report their results.

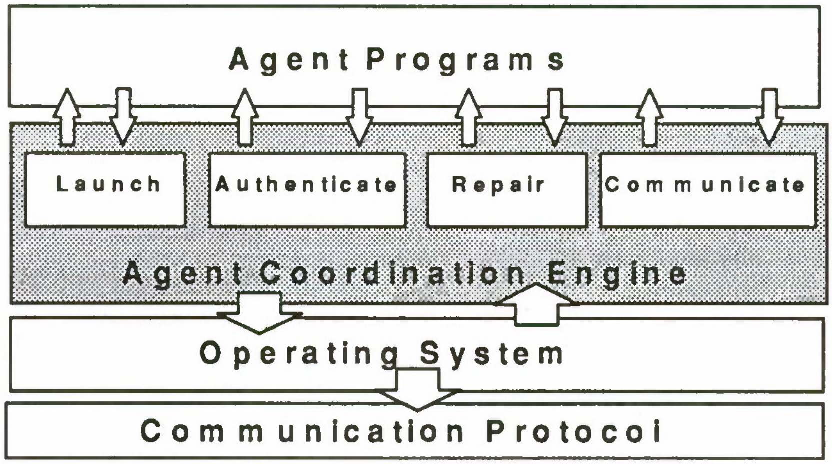


Figure 1 Centralized Agent Coordination

Launching Agents:
The coordination engine has the ability to launch or release agents into the computing environment. The engine will determine which, how many, and when agents are released into the environment. When the agent is launched or released, it is the responsibility of the ACE to ensure that the agent is informed of its scope and boundaries. The engine also verifies that the agents do not exceed their designated limitations.

Authenticating Agents:
In order to assure that the agents are performing the tasks they were designed and intended to perform, the coordination engine must ensure and verify the integrity of the agents used in the computing environment. Authenticating the agents consists of checking the state of the current agent with a known version. This can be accomplished through the use of such things as encryption, hashing or checksums.

Repairing Agents: In conjunction with the integrity of the autonomous agents ensured through authentication, the need to repair or disable damaged agents is necessary. If an agent is found to be damaged (corrupted), the coordination engine removes the damaged agent from service and repairs or replaces it. The repair process consists of replacing the damaged agent with an authenticated version of the agent available to the engine. In extreme cases the engine can notify the administrator that the agent needs to be reloaded from the original software.

Communication Agents: Since the autonomous agents independently perform their tasks, the coordination engine must provide a mechanism to coordinate the use and results of the agent's tasks. The results of the tasks need to be compiled to determine any further action that may be required, such as the release of additional agents.

With the agent operating environment established, the ACE acts to control the flow and use of autonomous agents within a given system. Acting in this manner, the agent operating environment and ACE closely resembles a biological immune system for computer virus response. In conjunction with the "biologically inspired immune system" [7], the use of autonomous agents suggests a more mobile and robust simulation of the immune system. With each agent performing a separate task, it can be suggested that the agents, in fact, act as biologically inspired "antibodies" for the computer system.

Using Autonomous Agents for Automated Virus Response

In a simplified description of the biological immune system, the antibodies detect entities which are foreign to it. Once a foreign body is detected and identified, it is destroyed by one or more antibodies. Acting as antibodies for a computer, autonomous agents need to perform similar functions for computer virus response. These functions, if initially performed from a known clean environment, can proactively prevent a virus infection at its source. This greatly reduces the risk of mass infections or epidemics which are currently experienced in many corporate environments [10]. As noted previously, these functions include the duties shown in Figure 2. Each portion of the automated response is described as part of the agent functions.

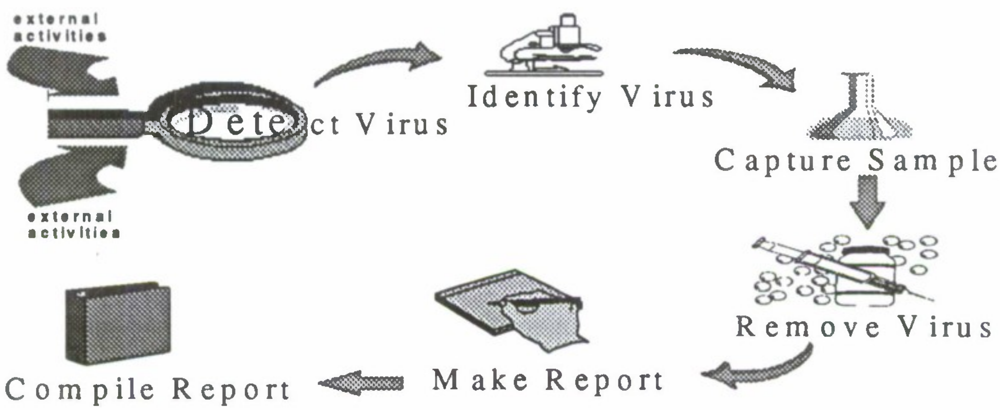


Figure 2 Automated Virus Response Duties

Detecting Viruses

To accomplish the detection of viruses, several autonomous agents are advisable to maintain the singular and simple task structure. Viruses come in three main forms: boot sector, file infector, and multi-partite. At a minimum, the automated response should include a separate agent for each type. Having separate agents for each type of virus allows the detection agents to continuously monitor different areas of the operating environment and to maintain the simple and singular tasks. In addition, each agent needs to be focused on a particular activity and can use different virus detection techniques. The current techniques for virus detection include scanning for known viruses using virus signatures, checking file integrity, and monitoring for suspicious behavior. The crucial activities for virus detection to monitor include:

- Inserting diskettes
- Receiving Mail
- Copying/Moving Files
- Creating/Saving Files

- Executing Files
- Opening Files

Once a virus is detected, the agent notifies the ACE along with the name/location of the suspected virus.

Identifying Viruses

Once a virus is detected (or suspected) using one of the virus detection techniques, agents must exist to positively identify the virus, if possible. In some cases, the detection agent may have a tentative identification; however, some of the detection techniques only detect a change, not the cause of the change. Again, to keep the agent task singular and simple, the identification of the virus is described separately from the detection. The identification of the virus is imperative to ensure proper recovery techniques are used. Since the focus of the automated response is on ITW viruses, the virus can be identified through either known virus signatures or known behaviors. Again, separate autonomous agents are advisable to identify boot sector, file infector and multi-partite viruses. The duties of the identification agents also need to be separate for each of the detection techniques used. There should be agents that handle viruses detected by known virus signatures, viruses detected by integrity checking and viruses detected by suspicious behavior. Once the virus is identified, the identity is returned to the ACE for appropriate recovery techniques. In addition, the identification agents are equipped to notify the ACE when the detected virus cannot be identified and, again, the ACE initiates the appropriate action(s).

Capturing Samples

Once the virus is detected and potentially identified, the ACE launches the appropriate agent(s) to collect a sample of the virus. Each capturing agent is supplied with the name/location of the infected item. Again, there is a separate agent to handle capturing boot sector, file infector and multi-partite viruses, since the tasks associated with each sample are different. To capture a sample, the agent makes a copy of the infected item and places it in a designated, protected location. A pointer to that location is sent to the ACE and the appropriate recovery agent is launched. For an unknown virus, the capturing agent activity is the same; however, the response from the ACE does not include a removal process, rather, it initiates the reporting agent(s).

Removing Viruses

After the sample is taken for ITW viruses, the ACE launches the appropriate agent for removing the virus. The information provided to the agent includes the name/location of the infected item and the identity of the virus. The recovery agent then determines the appropriate recovery technique for the identified virus and performs the necessary actions. Once completed, the recovery agent determines if the removal was successful and notifies the ACE of the removal status. If it was not successful, the agent notifies the ACE for appropriate reporting to the administrator.

Reporting Incidents

Once a virus is removed or, at least, the sample is taken (in the case of an unknown virus or unsuccessful removal), the ACE launches the reporting agent. The reporting agent generates a report of the incident including the date of the incident, the type of virus, the name of the virus detected and identified (if known), the location of the infection, and the success of the removal process, and other relevant information determined throughout the response. The agent then sends the report and the sample retrieved from the designated location to the administrator. The agent also sends the report and location of the sample to the repository site for future report compilation. Once the report is sent to the administrator and integrated into the repository, the reporting agent returns a completion notice to the ACE.

Compiling Reports

After reports are received from the reporting agent(s), they are stored in a repository site. The compiling agent(s) are launched to compile and generate reports. The agent may generate statistics based upon learned preferences [9] of the administrator. The compiled reports act as summaries of virus incidents and can be based upon specific intervals (i.e. monthly), virus type, virus name, or total incidents.

Future Considerations

There are many advantages for using autonomous agent technology, such as the ability of the agents to be easily tailored and trained, the efficiency, extensibility, scalability and graceful degradation of the agents, and the overall system's resilience to subversion [1]. While the advantages are numerous, there are also other considerations which will influence the use of autonomous agent technology for automated virus response. These considerations include: reducing processing overhead for the system, preventing deliberate or unintentional misuse, maintaining the integrity of agents, identifying the appropriate viruses to be included in an automated response, and providing accurate and consistent virus identification and recovery information. These considerations will impact the future directions taken for research in this area.

Reducing the Processing Overhead

While the agents themselves can be optimized to have minimal impact on system processing, the total automated virus response can impose an overhead on the computing system. The automated response will consume both memory and central processing time detecting and recovering from virus incidents. The use of memory and processing time will need to be minimized as much as possible to ensure that the benefits for automating the virus response are practical and can be realized. If the overhead imposed by an automated response degrades the overall performance of the system, the user community will disable or not install the product. The goal is not to decrease productivity but to enhance it.

Preventing the Misuse of Agents

Since agents can be defined as "good viruses" and have the ability to be executed throughout a system without user interaction or notification, it is imperative to ensure that the agent cannot be used for deliberate or unintentional misuse. Mechanisms will be needed to control the functions available to the agents and the scope or extent to which an agent can travel or perform its tasks. For instance, if an automated response is developed for a networked or client/server environment, the agents must be prevented from exceeding the boundaries of that environment. In addition, the system functions available to agents must be limited to those which do not allow the modification of other programs [5]. This can prevent an agent from being used to propagate viruses throughout the system or from changing programs to include Trojan horses.

Maintaining the Agent Integrity

As with the prevention of misuse, the integrity of the agents must also be ensured. Agents can be corrupted through deliberate or unintentional means. The results from executing a corrupted agent whether by design or accident can have disastrous results, such as system failure and data loss. It is possible to protect the integrity of the agents and the coordination engine with various forms of authentication or encryption. A possible method to protect the agent operating environment is to provide for integrity controls, such as authentication, through the design and implementation of a security architecture [11]. The mechanisms needed to maintain the integrity of the agents and their environment requires careful consideration to prevent a single agent or system of agents from causing harm.

Identifying the Target Response

Given that a small percentage of the viruses that exist are seen in actual incidents or in the wild, the automation of the virus response needs to focus its efforts on the detection and removal of the ITW viruses. To ensure that the automated response addresses the ITW viruses, a consistent designation of those viruses must be maintained and used. The *Wildlist* [14], maintained by Joe Wells of the IBM's T. J. Watson Research Center, provides a list of the viruses reported in actual virus incidents throughout the world. This list is currently being used by NCSA to test and certify anti-virus products [4]. The difficulties with the *Wildlist* are that the viruses noted as being in the wild currently contain naming variations and not all viruses actually in the wild are identified. Work is being done to address these issues [15]. Once the *Wildlist* and virus naming conventions are standardized, the targets of an automated response can be more clearly delineated.

Providing the Identification Information and Recovery Techniques

To minimize the impact of any virus response, it is important to have timely and accurate information on the identification and recovery of the ITW viruses. Accurate identification of viruses is important, since it directly affects the recovery process. It is the identification of the virus that determines the type and extent of the automated recovery process used. It is also imperative that the recovery techniques used for the ITW viruses are accurate and successful. Without successful recovery, an automated response loses its effectiveness and actually impedes productivity and fosters a false sense of protection. The fewer times that an administrator is involved with the recovery process, the fewer interruptions will be experienced by the user. Again, as in the identification of the virus, the recovery response needs to be standardized and robust enough to handle the ITW viruses consistently and effectively. It is possible that the agents could be trained [1,9] to determine the most appropriate recovery process if there are multiple infections present at the same time. In addition, false alarms are costly. In one case study, the cost of a small incident involving one virus and nine computers exceeded $23,000 in labor charges for lost time and productivity [12]. In actuality, the costs experienced in this case study were not significantly different than the costs that would have been experienced had the incident been real.

Summary

It is evident that the issue of computer viruses will be not going away in the near future. The current mechanisms for detecting and recovering from the growing number and complexity of computer viruses are no longer practical, timely, or efficient in regard to user productivity. The costs of training users and lost productivity due to virus incidents continue to rise as the complexity of both the operating environments and computer viruses increase.

Fully automating the response for the prevalent set of viruses would provide a considerable cost savings by eliminating the need for extensive training on the use of anti-virus products for the user and by reducing or eliminating user productivity interruptions. The generation of an immune system for computers using autonomous agent technology to combat virus infections can provide the automated response for computer viruses. Such an immune system can prevent the infection at its source by detecting a virus before it infects the computer or network. While the use of an automated response can be realized for known viruses with known recovery techniques, it should be noted that a fully automated response cannot be used in all computer virus incidents. For those incidents dealing with previously unknown viruses, expert technical assistance will still be needed.

The value of combining autonomous agent technology and automated virus response as suggested in this paper will be determined by the successful implementation of a prototype and operational use of the resulting automated virus response system. While researching and developing this prototype, the lessons learned throughout will be noted and used in determining other considerations, future directions and later versions.

The potential harm caused by making autonomous agent technology available for automated virus response provides a point to ponder. Are we providing the virus writers with a streamlined vehicle for virus propagation? As with most innovative concepts, autonomous agent technology can be used for both good and "evil". Arguably, autonomous agent technology can be readily seen as a threat, particularly in the virus arena. The challenge is to harness this advantageous but volatile technology to protect the computing environment from its most prevalent enemy, the computer virus [2].

Acknowledgments

I wish to thank the numerous people who provided me with support and encouragement to pursue this effort as well as providing useful suggestions and critical comments during the preparation of this paper. The initial idea for an autonomous, mobile immune system for computers grew out of a casual discussion with Bob Williamson, formerly of The MITRE Corporation, after a virus incident. I also wish to thank Mike Lambert, Frontier Corporation, whose encouragement and frank discussions about the methods and reasons for fully automating virus response brought this idea and resulting paper to fruition.

References

[1] Crosbie, Mark and Eugene H. Spafford, "Defending a Computer System using Autonomous Agents," *Making Security Real - 18th National Information Systems Security Conference Proceedings*, Baltimore, MD, October 1995, pp. 549-558.

[2] Datapro Information Service Group, "Computer Security Issues: 1995 Survey," McGraw-Hill, Incorporated, Delran, NJ, October 1995.

[3] D'haeseleer, Patrik, Stephanie Forrest, and Paul Helman, "An Immunological Approach to Change Detection: Algorithms, Analysis, and Implications," *Proceedings of the 1996 IEEE Symposium on Security and Privacy*, Oakland, CA, May 1996, pp. 110-119.

[4] Ford, Richard, "Why Viruses *Are* and *Always* will be a Problem," *NCSA News*, April 1996, pp. 5-7.

[5] Harrison, Colin G., David M. Chess, and Aaron Kershenbaum, "Mobile Agents: Are They a Good Idea?" IBM T. J. Watson Research Center, Yorktown Heights, NY, March 28, 1995, http://www.research.ibm.com/massive/mobag/ps.ps

[6] Heilmann, Kathryn, Dan Kihanya, Alastair Light, Paul Musembwa, "Intelligent Agents: A Technology and Business Application Analysis," November 1995. URL: http://haas.berkeley.edu/~heilmann/agents/

[7] Kephart, Jeffrey O., "A Biologically Inspired Immune System for Computers," High Integrity Laboratory, IBM Thomas J. Watson Research Center, Yorktown Heights, NY, 1994, gopher://index.almaden.ibm.com:70/0VIRUS/PAPERS/ALIFE.PS

[8] Lambert, Michael, "Fully Automated Response for In The Wild Viruses (FAR-ITW)", Rochester, NY, July 1995.

[9] Maes, Pattie, "Intelligent Software," *Scientific American*, volume 273, number 3, September 1995, pp. 84-86.

[10] National Computer Security Association (NCSA), "1996 Computer Virus Prevalence Survey," NCSA, Carlisle, PA, April 1996.

[11] Thirunavukkarasu, Chelliah, Tim Finin, and James Mayfield, "Secret Agents - A Security Architecture for the KQML Agent Communication Language," University of Maryland Baltimore County, Baltimore, MD, 1995, http://www.cs.umbc.edu/kqml/papers/secret.ps

[12] Trently, Christine M., "False Alarms: A Case Study" *InfoSecurity News*, volume 7, number 2, March/April 1996, p. 47.

[13] Wayner, Peter, *Agents Unleashed, A Public Domain Look at Agent Technology*, AP Professional, Chestnut Hill, MA, 1995.

[14] Wells, Joe, "PC Viruses in the Wild - February 10, 1996 (The *Wildlist*)," 1996.

[15] Wells, Joe, "Reality Check: Stalking the Wild Virus," NCSA's International Virus Prevention Conference Proceedings, April 1996, pp. Q1-Q13.

[16] White, Steve R., Jeffrey O. Kephart, and David M. Chess, "Computer Viruses: A Global Perspective," *Virus Bulletin Conference Proceedings*, Oxfordshire, England, September 1995, pp. 165-182.

Computer Virus Response Using Autonomous Agent Technology

Computer Virus Response Using Autonomous Agent Technology

Changing the Paradigm

Christine M. Trently
ctrently@mitretek.org

Outline

- Brief Perspective on Computer Virus Response
- Overview of Autonomous Agent Technology
- Agents for Computer Virus Response
- Example Using Agents for Virus Response
- Comparison - Current versus Future
- Future Considerations
- Conclusions

Brief Perspective - Computer Virus Response

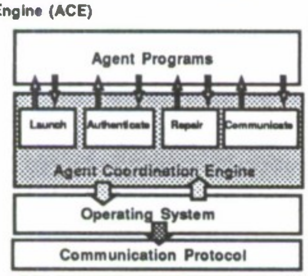
- Viruses ?
- Current Response
- Trends
- Need for Change
- Automated Response

Autonomous Agent (AA) Technology

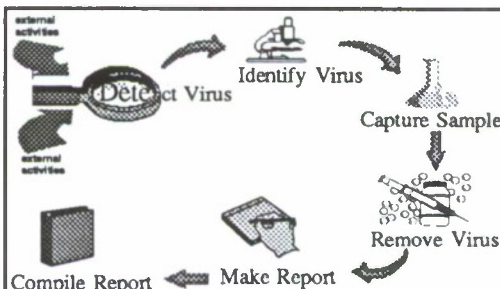
- Agent Characteristics
  - Simple, singular task
  - Mobile
  - Intelligence - Reasoning
  - Cooperation
- Operating Environment

Autonomous Agent (AA) Technology (concluded)

- Agent Coordination Engine (ACE)
  - Launch
  - Authenticate
  - Repair
  - Communicate



The Agents of Virus Response



Computer Virus Response Using Autonomous Agent Technology

Responding to Boot Sector Virus Using Agents

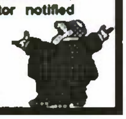
- Detect
  - Trigger: Insertion of diskette
  - Activity: Check for boot sector virus on diskette
  - Notification: Virus found message sent to ACE
- Identify
  - Trigger: ACE
  - Activity: Identify virus detected or Verify (virus) signature from detection
  - Notification: Virus identification to ACE
- Sample
  - Trigger: ACE
  - Activity: Make copy of Boot sector / disk image
  - Notification: Virus sample sent to repository and completion status sent to ACE

Response for Boot Sector Virus Using Agents (concluded)

- Recovery
  - Trigger: ACE
  - Activity: Remove virus from boot sector using appropriate technique
  - Notification: Completion status to ACE
- Report
  - Trigger: ACE
  - Activity: Generate incident report
  - Notification: 1) Report sent to administrator; 2) Report sent to repository 3) Completion status to ACE

Virus Response - Comparison Current vs. Future

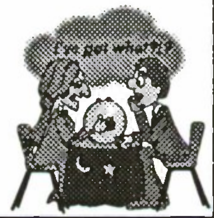
- Current (User Activity)
  - Scan computer periodically
  - Notified by AVS that BS virus detected
  - Boot from known clean, write-protected diskette
  - Take a sample by inserting new diskette
  - Use Recovery diskette or Run Clean-Up routine for given virus to remove virus
  - Re-scan
  - Return to Work
- Future (Agent Activity)
  - Activity on computer is checked by agents
  - BS virus found on diskette inserted into computer
  - Virus identified, if applicable
  - Sample taken
  - Virus removed
  - Report generated and administrator notified



Future Considerations and Conclusions

- Reduce Processing Overhead
- Prevent Misuse
- Maintain Agent Integrity
- Identify Target Response
- Provide Identification and Recovery Techniques

- Changing the Paradigm
- Providing a loaded gun



SECURITY ACROSS THE CURRICULUM: USING COMPUTER SECURITY TO TEACH COMPUTER SCIENCE PRINCIPLES

Major Gregory White, Ph.D.
Captain Gregory Nordstrom (ret.)
2354 Fairchild Dr., Suite 6K41
HQ USAFA/DFCS
USAF Academy, CO, 80840
white@cs.usafa.af.mil
gnordstr@cs.usafa.af.mil

ABSTRACT

Insuring that individuals who obtain computer science degrees have a sound foundation in security principles is becoming increasingly important as the worldwide connectivity of our networks grows and the number of security incidences increases. Increasing the number of courses a computer science major is required to take by adding additional computer science courses dealing with security is not the solution, however. Instead, an organized approach to include security topics into already existing curricula (as was first proposed in ACM's Curricula '91 document) is the key. This paper describes the approach taken at the United States Air Force Academy in introducing security topics at numerous points in its computer science curriculum. This approach goes far beyond briefly mentioning security at various points, pioneering the concept of using security to actually teach core computer science principles. This paper focuses in particular on changes that have been made to the Networks course required of all computer science majors which has been modified to use security to help illustrate and teach the underlying network principles.

INTRODUCTION

An ever growing number of colleges and universities have introduced courses in computer security. While this increased attention to security in academia is a good sign, the courses are being offered as elective courses. As an elective course, a significant number of students will not take these security courses which means that a significant number of computer science majors at these institutions will graduate without a solid background and basic understanding of security.

The ACM Curricula '91 document, proposed that a basic amount of computer security and ethics education be covered in all computer science programs. While the option to offer an elective course was acknowledged, the document proposed that a certain amount be covered at appropriate times in the curriculum. With the increasing need for computer professionals who have a solid grounding in security principles, this rather passive approach to security education is not sufficient. At the same time, computer science programs do not have the luxury of adding additional required courses to what in many cases is an already full program.

The solution to this dilemma is to introduce an organized approach to teaching security across the curriculum. Instead of addressing security topics as separate issues, security should be woven into all courses that make up the fabric of the core computer science curriculum. Indeed, what is needed is to make security considerations and concerns part of every programming assignment given to computer science students. In a manner similar to questions about good coding practices, students should be taught to always consider the security implications of any program developed.

The introduction of computer security across the curriculum should not come at the expense of other topics. Instead, security should enhance the learning of these other topics. Indeed, in certain courses, because of their very nature, security can actually be used to help teach the course itself. An example of this is a course in networks and computer communications which has numerous opportunities to introduce security related projects.

SECURITY ACROSS THE CURRICULUM

In today's heavily internetworked computing environment it is imperative that all students of computer science have an understanding of computer security principles and practices. Consequently, any implementation of security across the curriculum should begin with the first introductory computer science course. Many other majors today require some exposure to computers, in their introductory courses security should also be addressed. At this most basic level the detail required is minimal. Exposure to the concept of viruses and how to protect against them, good password management techniques, and elementary encryption issues will serve to introduce the students to the idea that security should always be a concern. Most of the time at this level is better spent in addressing the ethical and legal issues surrounding 'hacking' and viruses. Discussion on subjects like the ease in which electronic mail can be spoofed, or the fact that an individuals password or credit card numbers can be discovered using 'sniffers' will alert both the computer science major and the non-major alike to the real dangers that are present in placing too much trust in insecure networks. Programming assignments at this level will probably allow for few opportunities to address security concerns but research papers on subjects like public key encryption, malicious software, and 'hacking/cracking' provide ample opportunities to raise student's level of security awareness.

An operating system course provides many opportunities to address security issues both from a practical and a design point of view. Issues such as access control are already part of almost all textbooks on operating systems. Other issues such as authentication, object reuse, auditing, and security kernels also lend themselves to this course. For those interested in introducing even more security, the issues of multi-level security and its many additional requirements as well as the writing and detection of viruses and other forms of malicious software provide ample opportunities for programming projects.

While entire books have been written on data base security, many general textbooks designed for introductory data base courses often spend only a few pages on this subject or ignore it entirely. Issues such as multilevel protection, polyinstantiation, access modes, auditing, and inference controls provide a rich opportunity to reintroduce security concepts to the students.

Second only to operating systems in its opportunity to introduce security topics, a course in networks provides some of the best possibilities to stress the importance of security. This can easily be reinforced through the use of the many articles that appear in the news media concerning lapses in security protections in networks and computer systems. There are numerous security topics which can be used to illustrate or emphasize various network principles. Among these are cryptography, intrusion detection, firewalls, "worms", and security among distributed systems.

Software engineering courses with their emphasis on the entire life cycle of software also present several opportunities to discuss security issues. The design phase of software development provides the chance to discuss the modeling of secure systems. Discussion of program testing provides similar opportunities to discuss verification and validation. Covert channel analysis can also be easily introduced into this course.

USING SECURITY TO TEACH COMPUTER SCIENCE

The first course in which we attempted to use security to teach the principles embodied in the course was our senior level networks course. In the past, we taught the course centered around the seven-layer OSI model familiar to all who have taken an undergraduate-level network course. Lab assignments involved such tasks as development of programs to perform remote file transfer. These assignments, while providing examples of what was seen in lectures did nothing to motivate or excite the students. The labs were completed, the lessons learned, and the entire experience was then most likely quickly forgotten.

The most immediate benefit we observed using security to teach networking principles was a renewed enthusiasm for the course and computer science in general. Individuals who had been exhibiting only mediocre interest in their coursework came alive when challenged with our security related lab assignments.

The specific assignments used in this course began with simply downloading and running programs such as *crack*. This allowed the students to become comfortable with downloading and working with a program to get it to run on their specific system. It also served to illustrate how vulnerable a system is if an intruder is able to gain access to the password file. The students next learned to use the program *tcpdump* to monitor the packets that are sent across the network. Their assignment forced them to use several different options for this program and to track and observe many different types of packets that are sent across the network. When the assignment was distributed, we conducted a discussion on how this specific program, and other programs called 'sniffers', can be used to obtain passwords. The isolated nature of the lab meant the students weren't able to discover passwords to systems outside of their special subnet. While it would be absurd to assume that some student won't take advantage of this program on the isolated systems for mischievous purpose, the amount of damage, intentional or unintentional, that an individual can cause is very limited. This assignment also served to illustrate the different types of packets and their formats used in the TCP/IP protocol suite.

The next series of assignments had the students exploiting well known holes in a variety of packages. Many of these holes have been fixed in later releases of system software (which actually caused some problems as we had to insure that we didn't upgrade all of their systems). Examples of the types of holes/flaws they exploited include SMTP spoofing, the ***sendmail*** */etc/passwd* file hack, the TFTP */etc/passwd* file hack, and a ***uudecode*** spoof.

The culminating event for the course was the final project which was referred to as a 'hack-off.' For this assignment, the students were divided into teams which were further divided into two squads. Each team had an offensive and a defensive squad. The hack-off consisted of the teams attempting to break into their opponents systems while protecting their own. The systems they used were all on the isolated subnet and had been 'cleaned' prior to the event so they resembled their original, 'out-of-the-box' condition. The teams were provided a list of capabilities or functions their systems had to support at the start of the exercise. The instructors periodically checked the systems to insure the required capabilities still existed. This was done to insure that teams didn't simply "unplug" their system from the net and added a level of realism to the exercise. At various points in the exercise additional requirements were added to simulate the ever-changing environment administrators face. Not only did the students enjoy this project, they had the opportunity to actually get hands-on experience in minor system administration and security protection. The lessons they learned in this exercise will undoubtedly provide big dividends as they leave the academic environment.

ADDRESSING THE ISSUE OF 'HACKER' TRAINING

At first glance it may appear that the approach that we have taken at the Air Force Academy results in nothing more than a basic primer for the training of computer hackers. Implementation of a program similar to ours at other institutions where even less control of the students is possible will undoubtedly result in abuses of the information presented. During the initial implementation of this program, as the students and instructors were setting the boundaries, there were indeed minor incidents which were quickly resolved. Since these minor infractions, no problems have been encountered. We believe that this is partly due to the laboratory environment we have set up. We have a series of machines that were separated from the rest of our academic network which allowed the students to experiment in a controlled environment. Indeed, we encouraged them to test the security boundaries on these machines. Doing so has allowed our students to satisfy their curiosity and to learn many valuable security lessons without fear of destroying other important work in progress. At the same time, they could feel secure in that they did not have to hide their actions because of a fear of potential criminal prosecution. This fostered an environment in which the students freely shared the 'tricks' they learned.

We have had some claim that what we are doing is unleashing a new generation of trained hackers on the Internet. We do not agree with this sentiment. There are scores of hackers operating throughout the Internet today. We believe that hiding their techniques from our students only leads to a generation of system administrators who are 'sitting ducks' for the hackers that are out there. We use a knowledge of security holes to teach our students what must be done in order to secure their own systems. By doing so, our graduates are better able to handle the attacks on their systems that will surely occur.

CONCLUSIONS

As we have implemented our security across the curriculum program, we have noticed a number of benefits. The first one was a new level of interest in computer science from those who had previously not considered registering for the computer science major. There is a certain "frontier mystique" surrounding hackers and those who protect computer systems and networks from this new breed of "outlaws.'" On several occasions we have been able to use this interest to capture a student's interest long enough to explain the major to him/her which has resulted in an increase in the number of computer science majors.

Along with a new interest in the major, the introduction of security topics has renewed a number of the computer science majors interest in the program. A number of those, who had in the past shown less than total enthusiasm for the program, had a spark ignited in them with security and showed an improvement in their overall performance.

Using security to teach computer science principles did not detract from the other course material. We were able to use it to enhance the lessons being taught, to emphasize the points being made in a manner that the students found interesting. While this concept could be taken to the extreme and security forced upon all computer science courses, we did not take this approach, instead choosing to include it only in those programs for which we could see the course objectives easily applied to a security environment. This resulted in a well-balanced series of courses and an overall organized approach to applying the recommendations of the ACM Curricula '91 committee.

Finally, we entered into this experiment with a certain amount of apprehension surrounding the possibility that the things we taught could be used in an inappropriate manner. While we did indeed experience some minor incidents in the beginning, the students eventually settled down and did not push beyond the boundaries that were ultimately worked out. As a result, we do not believe that we have trained a corps of hackers, but rather have created a corps of "cyber defenders'" ready to leave academia and enter the work force prepared to defend their systems from the hackers that already, and will continue to, exist.

U.S. GOVERNMENT-WIDE INCIDENT RESPONSE CAPABILITY

Marianne Swanson
Computer Security Division
National Institute of Standards and Technology

Abstract

This paper describes the many functions that a federal incident response capability (IRC) would perform and explores the issues that should be addressed prior to the establishment of an IRC. The need for an incident response capability that crosses agency boundaries has never been greater. Almost all federal agencies are now connected to the Internet and exchange information regularly. The number of Internet related incidents that have occurred in the past year, along with the increase and complexity of viruses, requires agencies to take seriously their incident handling capability. The Office of Management and Budget has reinforced this need by requiring in the revision to OMB Circular A-130, Appendix III, that agencies be able to respond in a manner that both protects its own information and helps to protect the information of others who might be affected by the incident. A government-wide incident response capability (IRC) would assist civil agencies in meeting this requirement.

Introduction

The need for an incident response capability that crosses agency boundaries has never been greater. Almost all federal agencies are now connected to the Internet and exchange information regularly. The number of Internet related incidents [*figure 1.*] that have occurred in the past year, along with the increase and complexity of viruses, requires agencies to take seriously their incident handling capability. The Office of Management and Budget has reinforced this need by requiring in the revision to OMB Circular A-130, Appendix III, that agencies be able to respond in a manner that both protects its own information and helps to protect the information of others who might be affected by the incident. A government-wide incident response capability (IRC) would assist civil agencies in meeting this requirement. This paper describes the functions an IRC would perform and explores the issues that need to be addressed prior to the establishment of an IRC.

Background

The concept of a government-wide computer incident response capability has been researched and reported on since the Forum of Incident Response and Security Teams (FIRST)[1] was created in

---

[1]FIRST is an international group of incident response teams whose goal is to foster communication to prevent and to rapidly handle computer security related incidents.

1989. The original concept of how FIRST would coordinate the many incident response teams within thc FIRST organization, depict vendor teams, service provider teams, foreign govcrnment teams, U.S. military teams, several large U.S. federal teams, and one central U.S. federal team that would coordinate incident handling and information collection and dissemination for all other federal agencies. In the early 1990s, most agencies were not connected to the Internet and, except for cleaning up viruses, very few offered any formal incident handling support. The timing for the development of a government-wide IRC was too early.

CERT(sm) Coordination Center Statistics

Year	Incidents Reported(1)	Mail Messages Received	Requests Received(2)	Information Hotline Calls Received(3)
1988	6	539		
1989	132	2867		
1990	252	4448		
1991	406	9629		
1992	773	14463	275	1995
1993	1334	21267	1270	2282
1994	2341	29580	1527	3664
1995	2412	32084	1683	3428

Footnotes

(1) An incident may involve one site or hundreds or thousands of sites. Also, some incidents consist of ongoing activity for long periods of time (e.g., for more than a year).
(2) Information requests have been tabulated beginning July 1992. This number does not include requests to be added to mailing lists.
(3) Incoming hotline calls have been tabulated since January 1992. This number does not reflect total telephone activity related to incidents because outgoing calls made by CERT staff are not included.

Figure 1.

The IRC concept was again explored in 1993 by the National Security Telecommunications and Information Systems Security Committee (NSTISSC). This proposed plan for a national level IRC was limited in scope in that it would handle incidents affecting national security systems within military sites and civil sites. The report was presented to the NSTISSC member agencies in January 1993. The agencies agreed with the concept but could not support the resource commitment.

Recently, several other organizations have proposed an IRC concept. The General Services Administration, the National Communications System, and the National Institute of Standards and Technology have all prepared proposals to various funding bodies to obtain the much needed capital to seed such an enormous task. The outcome of these proposals -- whether they were approved, partially approved or turned down -- has yet to be determined. Clearly, there are many organizations that believe a national coordination of incident handling and a sharing of vulnerability related information is needed and needed soon

Scope

The IRC proposed by NIST is to provide a cost reimbursable incident handling service for those agencies not having sufficient resources to support their own capability. The IRC would facilitate the sharing of vulnerability information that would assist agencies in protecting their systems against known threats. The objective of the IRC would be to develop a self-sustaining incident response capability that meets the need of the federal agencies. Activities would range from providing agencies with direct technical support to handle computer security incidents, to providing backup support to agency response teams dealing with large and complex incidents, to providing agency response teams with information on threats, vulnerabilities, and countermeasures that allow agency teams to effectively deal with incidents on their own. Proposed activities include:

- Responding effectively and in a timely manner to security incidents: Coordinate the analysis of the problem, determine the magnitude of the threat, provide technical assistance in identifying and closing vulnerabilities, notify sites affected, and issue advisories to the agencies warning of the problem and describing countermeasures.

- Expanding the limited coverage of existing agency computer response teams by providing a broader range of incident types and technologies.

- Providing agencies with guidelines on implementing "fixes" and other security controls.

- Maintaining a 24 hour, 7 days a week response service for emergencies and a "Help Desk" function for normal business hours.

- Facilitating the interaction with law enforcement agencies in the reporting of security incidents involving violations of the law.

- Assisting federal law enforcement in evidence gathering, where appropriate.

- Coordinating with other organizations including FIRST.

- Developing, distributing, and maintaining publicly available security tools, incident handling tools and data gathering and reporting tools.

- Coordinating with vendors and Internet service providers to provide critical security patches and "work-arounds."

- Performing vulnerability analysis to identify a vulnerability's root-cause in order to identify other potential problems before they occur.

- Keeping the federal community aware of the current threat, i.e., education in current technology and associated threats; training of security and network administrators on security practices; and awareness through world wide web sites, ftp services and guidance documents.

The IRC would act as the central point of coordination and would establish channels to address incidents and vulnerabilities affecting agencies. A method for collecting, analyzing, and disseminating sanitized vulnerability, threat, and incident data would be developed. Activities in this area would include:

- Developing an acceptable use policy that defines the ways in which vulnerability data would be stored, protected, disseminated and used.

- On-going development of product vulnerability reports that describe product vulnerabilities along with known corrections, work-arounds, or countermeasures.

- On-going development of reports on intruder tools and techniques that describe methods of attack, potential impact, and countermeasures.

- Analyzing vulnerabilities to identify root-causes of problems in product development practices that produced the flaws and to support the development of tools that can test for other instances of similar flaws.

- Incident follow-up studies to identify the cause of the incident, operational impact on the affected organizations, and cost of resolving the incident, of recovering lost or damaged data, of restoring operation and of lost productivity.

Benefits

The primary benefits of this program would be:

- The immediate availability of the type of technical expertise and assistance that agencies need now to handle computer security incidents. The IRC will augment existing agency teams and provide assistance for agencies therefore, reducing the need to develop "full-function" incident response capability.

- The impact of security incidents will be contained and minimized by reducing the number of vulnerabilities among federal systems and by providing an early warning system that allows agencies to protect themselves from new threats.

- A centralized organization will review the nature of attacks to federal systems and will provide a common set of recommendations, tools and training to reduce the overall risk to federal systems.

Issues

To undertake a project with such far reaching goals, many questions must be answered. This paper does not attempt to answer them, rather the issues to each question are explored.

How to fund the capability? The biggest hurdle experienced so far is how to fund the IRC capability. Start up capital is required in order to be in a position to offer services immediately. Agencies need to get a return on their money and obtain the needed support; they are not in a position to wait six months or a year until the capability is staffed, trained and has the equipment to respond. If start up capital is secured, how to become self-sustaining is the next hurdle. All services would require an associated fee with possible plan options that agencies could buy into. For example, an agency may want to pay $25k for five days of incident handling support or $75k for a year of incident handling support for one firewall and all the systems connected to it. The fee structure should take into account all the functions the IRC would offer and price them competitively, yet reasonably enough for federal agencies to use them.

Who is responsible for the IRC? What government agency or Department is to be responsible for the federal IRC? GSA is in a position to contract out services; the IRC could be a service similar to the contingency planning hot-site agreements that currently exist through GSA's Federal Systems Integration and Management Center (FEDSIM). NIST could be considered a viable option for administering the contract and maintaining overall responsibility for its operation.

Who would operate the IRC? An existing team, like the Department of Energy's CIAC or like the CERT-CC, funded by DARPA, could take on the additional responsibility and workload and be ready to offer assistance immediately. By placing the IRC in an existing federal team, there would be no lag of six months or more until a new team is operational. The unique federal requirements would already be known. An argument can easily be made that a private incident handling team already in existence could be operational just as quickly and provide the same assistance as an existing federal team. The concept of placing the IRC within a federal agency and building it from the ground up should also be considered. By starting from scratch, the IRC can be built to exact specifications without the baggage brought in by an existing team.

What type of information should be handled? The NSTISSC report mentioned earlier described a need for a capability that would handle United States national security information. If a federal agency has an incident involving national security information, who does the agency go to for incident support? The DoD ASSIST team handles computer security incidents for military sites; does that include all classified incidents as well? Clearly, the IRC would need to work closely with the DoD teams to ensure that all national systems, including national security related systems, are supported if an incident occurs.

Conclusion

By having a centralized organization reviewing the nature of attacks, providing support, and sharing information, the security posture of federal systems are improved. The Administration recognized the need for incident handling and the sharing of incident and vulnerability data by establishing the requirement in the revision to OMB Circular A-130. With the requirement now in place, the time has finally come for a government-wide capability.

References

Culver, Grace. General Services Administration, *Draft Proposal: Federal Information Systems Security Incident Response Capability.* February 24, 1996.

CERT-Coordination Center, Software Engineering Institute. *Incident Statistics.* January, 1996.

Forum of Incident Response and Security Teams. *Operational Framework.* August, 1993.

National Institute of Standards and Technology. *Information Technology Fund Innovation Fund Pilot Program Proposal.* March 1, 1996.

National Security Telecommunications and Information Systems Security. *National Security Information Systems Incident Program (NSISIP).* January 1993.

MLS DBMS Interoperability Study*

Rae K. Burns, AGCS, Inc.
Yi-Fang Koh, Raytheon Electronic Systems

ESC/AXS, Building 1704, Hanscom AFB, MA 01731
burns/koh@stars1.hanscom.af.mil

Interoperability among heterogeneous databases is a fundamental requirement of many emerging Department of Defense (DoD) systems. Often these systems also have requirements for Multilevel-Secure (MLS) operation, where data is labeled to reflect its sensitivity level (e.g., UNCLASSIFIED, SECRET, etc.). The Air Force Rome Laboratory MLS Database Management System (DBMS) Interoperability Study has surveyed the available Commercial-Off-The-Shelf (COTS) products supporting interoperability and tested several of them in a multilevel environment. We selected representative products and implemented test scenarios in the ESC/AXS Security Products Transition Analysis Facility (STAF). Our test environment included three commercial MLS DBMS products (Trusted ORACLE7, Informix Online/Secure, and Sybase Secure SQL Server) on several different MLS Operating System (OS) platforms. We also employed "system high" platforms running standard versions of the DBMS and OS products. We successfully moved data to and from the MLS databases using different COTS interoperability solutions. This paper describes our testing efforts and summarizes the lessons learned.

1. Introduction

The Multilevel Secure Database Management System Interoperability Study was initiated by Air Force Rome Laboratory to expedite the transition of trusted database technology into operational Air Force C4I environments. The overall goal of the study is twofold:

1) To examine the theoretical basis for heterogeneous trusted database interoperability, identify issues, and transition findings to the communities involved in future development (vendors, DoD users, and applicable standards groups).

2) To develop demonstrable examples of database interoperability that illustrate the advantages of MLS DBMS products in support of Air Force operational requirements for multilevel security.

The results of the first goal were documented in the interim report [1] and are based on an analysis of current and proposed interoperability approaches documented in the database research literature. To address the second goal, a multilevel database testbed was created at the Security Products Transition Analysis Facility at Hanscom Air Force Base. Within the testbed, COTS database products have been installed, including both MLS DBMS products and connectivity products. The products have been integrated together using a simple Air Base Status database as

* This work was sponsored by Air Force Rome Laboratory under the USAF ESC/AXS PRISM Program, contract numbers F19628-92-C-0006 and F19628-92-C-0008.

the underlying application. This paper presents the results of our testing and demonstrations of INTERSOLV Open Database Connectivity (ODBC), Oracle PL/SQL Extender (PLEX), Sybase OmniSQL, and PRAXIS OmniReplicator. The operating system platforms in the testbed include SunOS 4.1.3, Sun Solaris 2.4, Sun Trusted Solaris 1.1, and Santa Cruz Operations (SCO) Secureware CMW+ 3.0. Standard versions of ORACLE7 and Sybase SQL Server are installed on SunOS platforms and the MLS versions on Sun Trusted Solaris platforms. Informix Online/Secure is installed on the SCO CMW platform. The connectivity products were installed on different platforms, depending on product availability and test scenario configurations.

1.1 Study Approach

During the course of the Interoperability Study, we analyzed and screened a number of different COTS connectivity products, and selected representative products to integrate into the STAF testbed. Based on our analysis we identified four categories of COTS interoperability products: standards-based, vendor-specific, gateway, and replicator.

(1) Standards-based Solutions

We looked at two different standards-based interoperability approaches: Open Database Connectivity (ODBC) and Remote Data Access (RDA). Microsoft's ODBC interface [2] is one of the first implementations of the SQL Access Group (SAG) Call Level Interface (CLI) standard. ODBC is based on the X/Open and SAG CLI 1992 specification [3], defining a C or C++ programming language interface for standardized DBMS connectivity. We successfully used two different INTERSOLV ODBC products to access data in an MLS database.

The International Standards Organization (ISO) Open Systems Interconnection (OSI) RDA standard [4] defines the message format for sending SQL queries to a DBMS and receiving data from the DBMS. The National Institute of Standards (NIST) has supported efforts to promote the standard, but currently there are few COTS products that implement the RDA standard. The major MLS DBMS vendors currently only support proprietary message formats and do not provide RDA interfaces to their products. Consequently, for the Interoperability Study, we did not perform any testing with RDA-compliant products.

(2) Vendor-specific Solutions

Several DBMS vendors support interfaces that facilitate interoperability but do not provide a general purpose solution. We used an Oracle Federal tool, Oracle PLEX, to integrate data from a Sybase Secure SQL Server database into a Trusted ORACLE7 application.

(3) Gateway Solutions

Gateway products generally map the SQL schema from one DBMS onto an equivalent schema in another DBMS, giving the user transparent read and/or write access to a foreign data source. We tested the Sybase OmniSQL gateway product in two different configurations. In one configuration, we retrieved data from two different MLS databases; in the other, we loaded data from multiple single-level databases into a central multilevel database.

(4) Replicator Solutions

Replication supports the automatic updating of remote databases based on changes made to another source database. Our experiments with PRAXIS OmniReplicator included replication among MLS databases as well as replication from a single-level database into a multilevel database.

1.2 Example Database

Since the scope of this study was limited to interoperability issues, we chose a simple application for testing each connectivity product. The Air Base Status database contains information about the facilities and runways of several air bases and indicates the current status of the base and each of its runways (e.g., whether it is operational or not). We created this database on each of the platforms in the testbed and populated it with a small amount of data on air bases in the Persian Gulf. We then designed tests for the interoperability products that retrieved and updated the status information using different operational scenarios.

1.3 Product and Configuration Limitations

Because of our focus on available COTS technology, we were limited in the level of assurance achievable in our test configurations. The MLS DBMS products we used have been evaluated (or are being evaluated) at B1. The Compartmented-Mode Workstation (CMW) platforms we employed are also fundamentally B1 class systems. We limited our test scenario accreditation range to CONFIDENTIAL and SECRET, with some releasability compartments, in order to demonstrate the MLS DBMS and connectivity products in an appropriate risk environment.

None of the COTS connectivity products we used have been evaluated nor were they targeted for use in a multilevel context. Their use imposes additional limitations since none of the products were able to directly interpret sensitivity labels. To retrieve sensitivity labels, we created views within the multilevel databases that automatically converted the sensitivity label to a character string. By accessing these views instead of the base tables, the sensitivity labels were made available to the connectivity products as *advisory* labels. (The labels can only be advisory since the COTS connectivity products are not *trusted* to manage sensitivity labels.) For database updates, a connectivity product was run as a single-level process; consequently, all updates were labeled by the MLS DBMS with the sensitivity label associated with that process.

The remainder of this report describes the COTS solutions and our experiences installing, configuring, and demonstrating them in the STAF multilevel database testbed.

2. Open Database Connectivity

The ODBC interface allows a user to write a single application to access databases managed by different DBMS products. SQL statements can be included directly in the source code or can be constructed dynamically at run time. The underlying communication with the DBMS is completely transparent to the application.

ODBC architecture includes four components as illustrated in Figure 1. An **Application** uses the **ODBC Application Programming Interface (API)** to call ODBC functions that submit SQL statements and retrieve data. The **Driver Manager** loads drivers on behalf of an application, then the **Driver** for a specific DBMS processes ODBC function calls and submits the SQL statements to the designated **Data Source**. We successfully integrated two ODBC products to access different data sources:

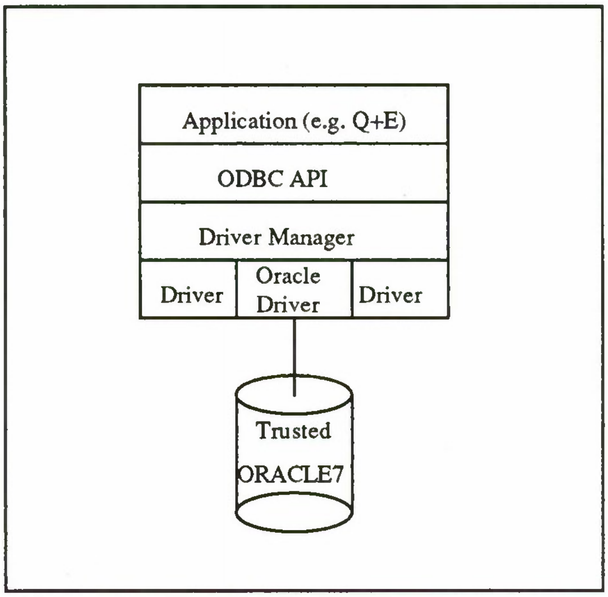


Figure 1. ODBC Configuration

INTERSOLV Q+E for Windows: Q+E is a query and reporting application that uses ODBC drivers to provide access to a large number of different database products [5]. We installed Q+E and the ODBC Driver Pack [6] on a Windows NT platform. We also installed Oracle SQL*Net for Windows to provide connectivity to Oracle databases. After configuring both the ODBC.INI file and the local Oracle configuration files to refer to the Trusted ORACLE7 database, we were able to retrieve information and generate customized Q+E reports that included both the data and sensitivity labels.

INTERSOLV ODBC for UNIX: The ODBC drivers for UNIX were installed on a Sun Solaris 2.4 system. The Oracle ODBC driver was linked with the standard ORACLE7 client libraries for Solaris. We then wrote a C program that used ODBC functions to connect to the Trusted ORACLE7 database on a Sun Trusted Solaris 1.1 CMW. The program simply retrieved Air Base Status information (including sensitivity labels) and displayed it interactively to the user. Since we did not have client libraries for any of the other MLS DBMS products on Sun Solaris, we did not test the other MLS databases.

The specific configuration parameters for the UNIX and Windows environments are documented in the ODBC Interoperability Report [7] along with the C source code developed for the UNIX testing. The ODBC API does provides a DBMS-independent interface that can be used to access MLS databases.

3. Oracle PLEX

Oracle PLEX is a vendor-specific interoperability solution that allows an Oracle application to access foreign data sources [8]. It provides a set of functions that extend the capability of Oracle's Programming Language/Structured Query Language (PL/SQL) to communicate with an *application server* that performs operations outside the scope of a traditional database application. The PLEX product provides a number of program development tools to build both the application server and the PL/SQL modules used to communicate with the server. For our interoperability test, we developed an application server that retrieved data from a Sybase Secure

SQL Server database and displayed it using Oracle PL/SQL routines in a Trusted ORACLE7 database. The application server used the Sybase DB-Library API to retrieve data from the Air Base Status database. While we were successful in accessing a remote database using PLEX, the solution was fairly complex and did not provide a generic interoperability solution, as documented in the Oracle PLEX Interoperability Report [9]. However, for access to non-database information, ORACLE PLEX provides a viable basis for using Oracle's PL/SQL, rather than C, for application development.

4. OmniSQL Gateway

The Sybase OmniSQL Server allows an application using the Sybase Open Client API to access databases managed by other DBMS products[10]. Information about the other databases is stored locally in an OmniSQL database as mappings from the local environment to the remote environments. Both user identifier mappings and table definition mappings are maintained by OmniSQL. When a local request is made, OmniSQL uses the mapping information to access the appropriate data sources.

There were two different test scenarios established for the OmniSQL interoperability testing. First, the OmniSQL Server was installed on a SunOS system and used to combine data from a Trusted ORACLE7 database and a Sybase Secure SQL Server database. In the second scenario, multiple instances of the OmniSQL Server were run on a Sun CMW at different sensitivity labels and used to update a central Sybase SQL Server database.

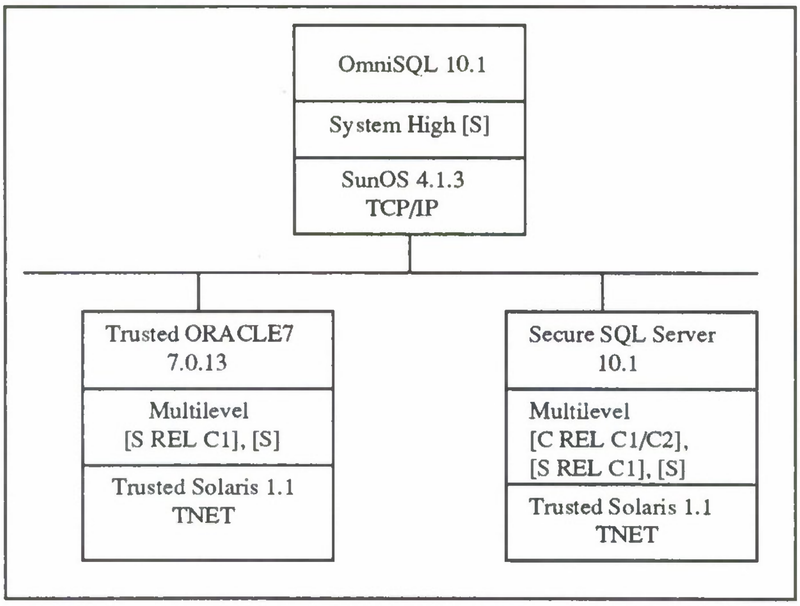


Figure 2. OmniSQL SunOS Environment

Figure 2 illustrates the configuration for the first scenario. To combine data from the Trusted ORACLE7 database with data from the Sybase Secure SQL Server database, we created a stored procedure in the OmniSQL database that referred to both of the MLS databases. We first had to map the MLS data (e.g., views with the sensitivity labels converted to character string datatypes) to locally defined OmniSQL tables. OmniSQL provides a utility to automatically generate the required mapping specification from the table definition in the remote database [11]. However, we had difficulty using the utility because it did not expect the extra sensitivity label column that the MLS DBMS products append to each table. (While Trusted ORACLE7 only appended the label column to base tables, Sybase Secure SQL Server appended it to views as well.) The problems we encountered are documented in the OmniSQL Interoperability Report [12]; however, we were able to successfully combine the data from the MLS databases after dealing with the sensitivity label problems.

For the second scenario, the OmniSQL Server software was installed on a Sun CMW. We set up multiple OmniSQL Servers to retrieve the Air Base Status data at three different sensitivity levels from three different databases (standard Sybase, Trusted ORACLE7 and ORACLE7) and update a central multilevel Sybase Secure SQL Server Air Base Status database.

Three different sensitivity labels were used in this testing: [C REL CNTRY1/CNTRY2], [S REL CNTRY1], and [S]. An OmniSQL database and a server instance were required at each sensitivity level in order to communicate with the remote database at a single level. In addition to the table mappings required for this scenario, we set up an OmniSQL stored procedure at each level. The stored procedure first retrieved the requested status information and then used that data to update the status information in the central multilevel database. Figure 3 illustrates the configuration for this scenario.

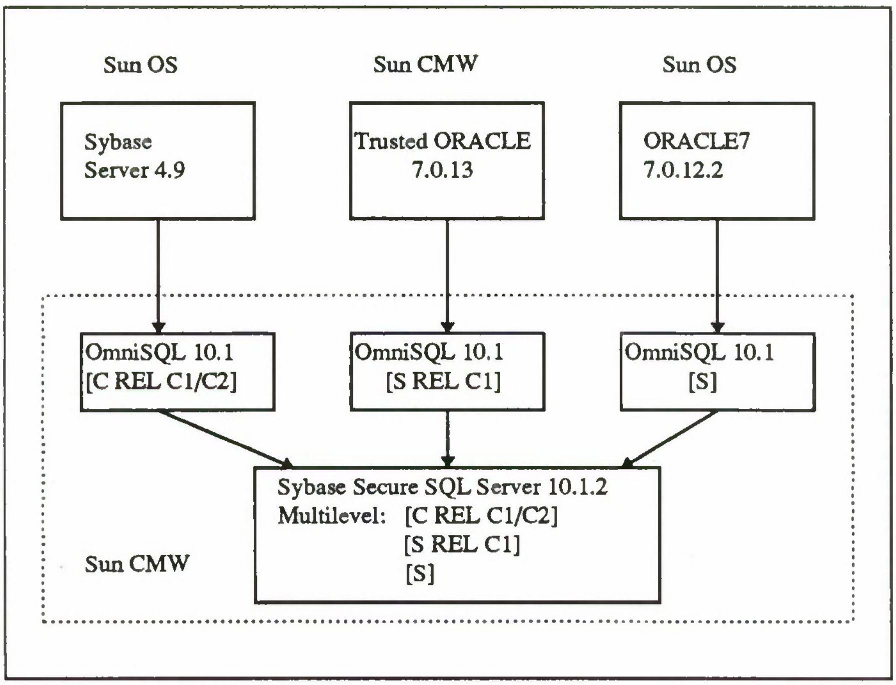


Figure 3. OmniSQL Multilevel CMW Environment

During the OmniSQL Server testing, several interoperability problems surfaced that involved the network configuration definitions on the Sun CMW platforms. The first problem was caused by the Trusted ORACLE7 SQL*Net listener process. The listener process executes with privilege and runs at the *lowest* level of data within the database. Our first scenario was accessing tables at the *highest* level from a single-level untrusted SunOS system. This caused several different problems, all of which were resolved by small modifications to the Sun CMW network host configuration file (TNETRHDB). We had similar problems with Sybase Secure SQL Server and floating information labels which were also solved by minor changes to the TNETRHDB file.

Details on these problems and their solutions are documented in the OmniSQL Interoperability Report [12].

Within both of these test environments, we were able to demonstrate that the OmniSQL Server can be used successfully to retrieve and update data from databases managed by dissimilar MLS DBMS products without violating the overall system security policy.

5. OmniReplicator

The PRAXIS OmniReplicator is a replication server designed to work with heterogeneous databases [13]. The tables and columns to be replicated are specified by an administrator using an OmniReplicator application on a PC. The administrator application connects to the source database, creates tables for use by the OmniReplicator, stores the replication configuration as defined by the administrator, and creates triggers to capture the updates to be replicated. When OmniReplicator processes run on the source platform, they uses the tables within the source database to control and monitor the replication activities. For communication to the target database, OmniReplicator relies on the SequeLink product from INTERSOLV. SequeLink transforms the update statements into messages and transmits them to the target database.

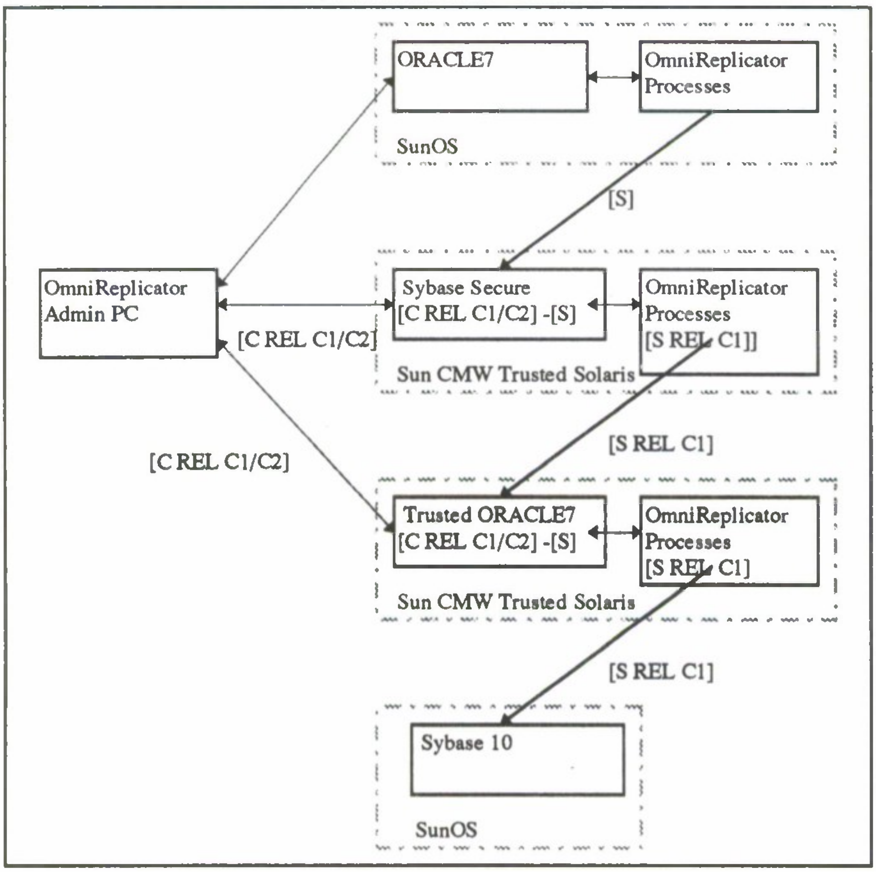


Figure 4. OmniReplicator Configuration

For our interoperability test, we set up a cascading configuration, as illustrated in Figure 4. The standard version of ORACLE7 is the source database. When updates are made to it, they are replicated to the Secure Sybase database at [S]. These updates to the Sybase Secure database cause replication to the Trusted ORACLE7 database. We used the Sybase *trusted trigger* feature to perform a downgrade from [S] to [S REL C1] prior to replicating the data to Trusted ORACLE7. We designated the OmniReplicator triggers as trusted and authorized them to "writedown" to [S REL C1] when they stored data. As indicated in Figure 4, the Sybase Secure OmniReplicator processes execute at [S REL C1], not at [S], to read and process the replication information stored by the trusted triggers. In turn, updates to the Trusted ORACLE7 database cause replication to a standard Sybase database.

The installation of the SequeLink and OmniReplicator software in a multilevel environment is fairly complex [14]. The SequeLink product is intended to be installed by the **root** user, however we modified the install script and installed it without privilege. In addition, when a remote user tries to connect to the SequeLink Server, SequeLink attempts to validate the user's login ID and password. However, on the CMW systems, the **/etc/passwd** file is not the same as on a regular UNIX system. The **/etc/passwd** file does not contain encrypted passwords, so SequeLink initially rejected remote connections. To get around this problem, we put an entry in the **/etc/passwd** file for the account we were using for our replication testing; this solved the problem without creating any errors with the CMW user authentication. Finally, since we needed to connect at several different labels, we created a multilevel directory to store the SequeLink log files.

For the Sybase version of SequeLink, we also had to modify the SequeLink stored procedures that retrieve datatype information. We removed the sensitivity datatype since the OmniReplicator Administrator software on the PC did not correctly interpret the sensitivity datatype. We also had to modify the scripts that create the OmniReplicator tables within the source Sybase database. These modifications were extensive, since the creation of multilevel tables in Sybase requires extra parameters on the CREATE TABLE statements. Finally, we had a problem with the use of the **sa_role** feature. The OmniReplicator software assumes there is a Sybase user, **rpdbo**, who is authorized for the sa_role. However, in Secure Sybase, the role must be *enabled* before it is effective, so the operations requiring the sa_role failed. We finally managed to overcome this problem by changing database ownership and privileges so that the use of the sa_role was unnecessary. We had fewer problems with Trusted ORACLE7, but still had to make a couple of changes to the installation scripts.

Once the replication parameters were correctly setup using the OmniReplicator Administrator, we were able to cascade the replication of Air Base Status information from the original source through the intermediate multilevel databases. The use of a trusted trigger within the Sybase Secure database supported an automated downgrade. Additional features of OmniReplicator could be employed to replicate data to different target databases based on data sensitivity labels and to sanitize data as it is replicated.

6. Lessons Learned

The overall conclusion of this Interoperability Study is that COTS connectivity products can successfully be used to support Air Force operational requirements in a multilevel environment. Since the COTS DBMS products currently do not provide *high assurance* solutions, the accreditation range for an operational system will necessarily be limited.

We encountered two fundamental problems with sensitivity labels. First, each of the MLS DBMS products define the sensitivity label column differently, both in column name and datatype. In general, COTS connectivity products do not recognize the sensitivity label datatype and cannot interpret it. An SQL standard for sensitivity labels would greatly facilitate interoperability using

COTS products. The standard would primarily need to address the label *datatype* and how the label appears to client applications (e.g., as a character string or as an internal label to be interpreted by the client software).

A second problem with sensitivity labels is that they can only be advisory. If there were an appropriate infrastructure established to deliver a sensitivity label along with the data to a *trusted* component on a client system, then a trusted application could display the data and sensitivity labels. The infrastructure could involve extensions to trusted networking (e.g., a security API as proposed by the Trusted Systems Interoperability Group (TSIG)) or could be based on digital signatures (e.g., where the MLS DBMS would sign both the data and its label before returning it to a client application). In order to have trusted sensitivity labels, some additional infrastructure must be developed.

The other problems we encountered had to do with the trusted networking parameters and the security environment on the MLS platforms. Configuring a single MLS platform is complex and difficult. To correctly configure a heterogeneous MLS network of any size is even more difficult. Our problems with the Sun CMW network configuration parameters were the result of some subtleties involved with privileged software accessing single-level hosts. Further standardization of multilevel network protocols, configuration parameters, and label translation capabilities would substantially improve the administrator's ability to configure a secure heterogeneous network that supports database interoperability.

Future work in MLS database interoperability should address both the sensitivity label issues and the trusted networking infrastructure. In addition, as higher assurance operating systems and database management systems are developed, interoperability using high assurance MLS database servers should be pursued. A high assurance database server could support a wider accreditation range and could be accessed from both system high platforms and low assurance B1 and CMW platforms.

7. References

1 Interoperability of Trusted, Heterogeneous, Autonomous, Distributed (THAD) DBMS Interim Report, Security Products Transition Analysis Facility, ESC/AXS, Hanscom AFB, 2/22/95.

2 Microsoft ODBC 2.0 Programmer's Reference and Software Development Kit (SDK) Guide, Microsoft Press, 1994.

3 X/Open CAE Specification, Structured Query Language (SQL), (ISBN:1-872630-58-8 C201), August 1992,.

4 X/Open CAE Specification: SQL Remote Data Access, (ISBN: 1-872630-98-7 C307), 1994.

5 Q+E User's Manual, INTERSOLV, January 1995.

6 Data Direct ODBC Drivers, INTERSOLV, February 1995.

7 ODBC Interoperability Report, Security Products Transition Analysis Facility, ESC/AXS, Hanscom AFB, February, 1996.

8 Oracle PLEX Installation and User's Guide, Oracle Federal, August, 1994.

9 Oracle PLEX Interoperability Report, Security Products Transition Analysis Facility, ESC/AXS, Hanscom AFB, February 1996.

10 OmniSQL Server System Administration Guide, Sybase Inc., March, 1993.

11 OmniSQL Server Class and Utilities Reference Manual, Sybase Inc., November 1993.

12 OmniSQL Interoperability Report, Security Products Transition Analysis Facility, ESC/AXS, Hanscom AFB, February, 1996.

13 OmniReplicator Concepts and Facilities, Praxis International, Inc., 1995.

14 OmniReplicator Interoperability Report, Security Products Transition Analysis Facility, ESC/AXS, Hanscom AFB, July 1996.

MISSI Compliance for Commercial-Off-The-Shelf Firewalls

Michael Hale
Tammy Mannarino
National Security Agency
Network Security Group
9800 Savage Road
Fort George G. Meade, Maryland 20755-6000

Phone: (410) 859-4471

e-mail:
tlmanna@missi.ncsc.mil
mwhale@missi.ncsc.mil

Abstract

The Multilevel Information System Security Initiative is a framework for complete information security in computer networks. The Network Security Group is defining commercial-off-the-shelf network security solutions that fit into this framework. Internet firewalls must adhere to the security and interoperability requirements for the Multilevel Information System Security Initiative. The Network Security Group will evaluate and test candidate products against these requirements to give objective information about the products to DoD Services and Agencies.

1.0 Multilevel Information System Security Initiative

The purpose of the Multilevel Information System Security Initiative (MISSI) is to define an open, distributed security framework for the Defense Information System Infrastructure (DII). It is an initiative of NSA's Information Systems Security Organization (ISSO), which is responsible for providing technical guidance and solutions to the network security problems of U.S. Government Classified and Unclassified National Security-related systems. MISSI is intended to make solutions which provide secure interoperability available to users with a wide variety of needs. Commercial hardware and software products will be used within this framework. Firewalls are one of the "building block" products of the MISSI architecture. Firewalls fall into the category of system/enclave security products which also includes guards and in-line network encryptors. System/enclave security products generally reside at the local enclave boundary and provide access control and/or encryption services between the enclave and external networks.

Fortezza is a workstation security product which is also a MISSI building block. The Fortezza card, a PCMCIA card, when used with its associated personal identification number (PIN) provides access to Fortezza secured applications such as e-mail, file transfer, World-Wide Web browsers, remote database access, file storage, electronic commerce/electronic data interchange and remote identification and authentication. Fortezza provides protection via identification and authentication, confidentiality, data integrity and non-repudiation services for sensitive but unclassified (SBU) data.

There are more than 40 commercial-of the-shelf (COTS) firewalls. These firewalls rely on a variety of techniques for their claimed security capabilities. It is the intention of the MISSI program to take advantage of the wealth of commercial products in the market to provide the security and interoperability required by the DII. The large number of products and the different approaches of each brings with it confusion for customers. They need to determine which firewalls meet a minimum threshold for security and also which firewalls are interoperable with other MISSI components

MISSI Compliance for COTS Firewalls grows out of the need to eliminate this customer confusion. The activity consists of 1) defining a minimal set of security and interoperability requirements for all firewalls to be used in the DII environment, 2) communicating these requirements to firewall vendors and firewall customers alike and 3) testing commercial-off- the-shelf firewalls to determine whether they meet these requirements.

There is a great deal of debate concerning the difference between Firewalls and Guards. Currently, the MISSI program is working to specify the differences in features as well as assurances. NSA believes using firewalls (products of lower assurance) for protecting unclassified or sensitive but unclassified information is acceptable, but that a higher assurance guard is required when protecting classified networks. Functional differences between firewalls and guards are still being debated. Since there are differences between firewalls and guards, a separate compliance program for each is being developed. The security requirements for guards (particularly the assurance requirements) will be much more stringent than for firewalls. MISSI compliance requirements for guards will be developed by mid-1996.

2.0 Challenges

One of the obstacles to creating a minimum threshold for security is the fact that the firewall market is evolving rapidly. New capabilities are being continually added. Users are demanding more services and greater interoperability of firewalls while demanding the same or an increased level of security. Any set of security measures, criteria or requirements will have to grow and change with the technology, with the way consumers are using firewalls and the environments in which they are being used. The MISSI requirements are no exception; they will need to be updated on a regular basis in order to keep pace with the state-of the-practice.

Firewall customers are constantly asking the same question "which one is the 'best' firewall?" This question is difficult to answer because it depends almost entirely on the way in which the user intends to implement the firewall. Some key factors that should be weighed in the selection of a firewall are: the value of the information to be protected, the identity and skill level of your adversaries, the services your organization wishes to use through the firewall, and whether users will access the protected network from outside. Since everyone's needs may be different, it is difficult to state a set of requirements that will apply to all users of firewalls. It is for this reason that we have chosen to break down the MISSI compliance requirements by environment. We are creating a set of requirements for sensitive, but unclassified environments and one for secret environments. In general both sets of requirements generally describe or apply to a large homogeneous group. We recognize that within this large group there will be some variation, which is why in addition to a minimum set of requirements there are optional requirements which will provide additional security at the discretion of the user.

Yet another difficulty of testing is how to measure firewalls that use different approaches to meet the same set of requirements. Given that there are packet filtering firewalls and application gateways and firewalls that use type enforcement, we must ensure that the requirements treat the different types of products even-handedly. The requirements are written in implementation-independent language. They focus on what the firewall should do rather than how it should do it.

3.0 Current Firewall Testing Efforts

There are a number of testing efforts being carried out in the firewall community. One way of distinguishing the scope of these efforts is to separate product testing from system testing. Product testing refers to the testing of a commercial firewall in a generic configuration. Many firewalls can be tested in the same generic test bed in order to obtain an "objective" comparison of firewall capabilities. System testing, on the other hand, is the testing of a firewall within a specific system environment, usually the configuration (or a simulation) in which it is to be used. This arrangement allows testers to configure the firewall as it would be used, and to test it in a true operational environment.

Both of these types of firewall testing are important and are currently being planned and performed. The organizations performing these types of tests differ. Product tests are being performed by third party organizations who are not involved in the business of buying or selling firewalls. System tests are usually run by the purchasers (or prospective purchasers) of a firewall or by consultants hired by the purchasers. Since the number of organizations purchasing firewalls is very large, the variety of system testing methodologies is great and therefore hard to categorize. This paper will instead describe some product testing efforts whose results will apply to a greater audience. For each testing activity, we will state the sponsor, purpose and audience and provide additional information where appropriate.

The first type of product testing or evaluation we will discuss falls under the aegis of the common criteria effort sponsored by organizations in the US, Canada, UK, France, Germany and Denmark) [3]. Under this program a Common Criteria Protection Profile (CCPP) for packet filtering firewalls has been developed and a protection profile for application gateway firewalls is in progress. The packet filtering firewall profile currently documents firewall market practices. With the exception of some additional auditing requirements all of the requirements could be met by most commercial firewalls available today. The packet filtering requirements are written to the AL-1 assurance level. The intent of the profile is for evaluators to be able to test and document that a particular firewall meets that level of assurance. The profile requires conformance testing by the Vendor, validation of that testing by evaluators and testing of the firewall for obvious flaws. It is written in terminology using the Orange Book concepts of subject and object. The packet filtering firewall profile has been in draft form since June 1995. By the time this paper is published a final profile will be delivered.

The Air Force Information Warfare Center (AFIWC) is another group performing firewall testing. They have completed what they term a "quick look" test of the Sterling Connect: Firewall and plan to test more commercial firewalls in 1996. One of the main differences between this testing and that of the CCPP is that AFIWC has no fixed set of requirements to guide their testing. Their testing validates vendor claims and includes a significant amount of penetration testing. The audience for this testing is for the most part the Air Force, DoD and the U.S. government but the test reports are unclassified and can be used by anyone. NSA has also been performing this type of testing with the same basic purpose and audience. NSA has tested Sidewinder version 1.0 and is in the process of testing the Gauntlet Firewall.

In NSA's MISSI compliance program, the requirements are written as a minimum essential set of what is needed by DoD SBU enclaves. They are a target for commercial firewalls. Most of the firewalls on the market will not yet have the stated functionality. The MISSI requirements, in addition to having assurance and management requirements, have separated the remaining requirements into message-oriented and session oriented protocols. In contrast, the CCPP requirements are protocol-independent in the packet filtering profile. The audience for the MISSI compliance requirements has two main groups: 1)the evaluators who will use the requirements to test a commercial firewall for compliance and 2) systems engineers who will use these requirements as a baseline in developing their own set of requirements for a firewall they will acquire or design. NSA began to develop these draft requirements in the summer of 1995. A final set of requirements was released in early 1996. The first compliance tests will be performed beginning in May 1996.

4.0 Security Requirements for MISSI Compliant Firewalls

The security requirements for MISSI compliant firewalls [2] are written with the intended fielded environment in mind. The characteristics of the environment are crucial to shaping the security requirements. For MISSI compliant firewalls, one can assume that the environment processes information of considerable sensitivity, but the information has no formal security classification. This is usually referred to as sensitive but unclassified. The firewall protects the computers in the SBU environment from hostile or unauthorized access originating from other locations on the Internet.

Many local security policy parameters may affect the configuration of the firewall. One such parameter is the designation of which computers outside the local SBU environment are allowable communication partners. One possibility is that the local policy constrains the computers in the SBU environment to communication only with other designated computers that are in other SBU environments. This is referred to as SBU-SBU. The opposite possibility is to not limit the set of communication partners, thereby allowing communication with any and all computers on the Defense Information System Network and the Internet.

MISSI is a complete network security solution, with many components providing various aspects of the overall network security. Assumptions that these other components will be in place eliminate certain requirements for the firewall. For example, the firewall requirements do not dictate that the firewall provide the features of encryption and digital signatures on the network traffic passing across the enclave boundary, although some firewalls do provide this service. In the ultimate vision for MISSI, Fortezza cards at the end workstations encrypt/decrypt and sign/verify as necessary. The firewall provides complementary security functions.

The requirements are divided into two basic sets: those for session-oriented protocols and those for message-oriented protocols. Session oriented protocols are those protocols that provide a stateful, enduring communication session between a client and a server. Such a session allows data to flow in both directions. Examples are *ftp* and *telnet*. Message-oriented protocols, on the other hand, are stateless and non-enduring. One can send an e-mail message, and a response may or may not arrive from the other end. The message lasts of no duration in time, nor does it have an open or closed state as a session would. The requirements for session-oriented protocols and message-oriented protocols are discussed in sections 4.1 and 4.2, respectively.

4.1 Session Oriented Protocols

A MISSI compliant firewall protects servers from access by external clients that are not authorized to establish sessions. Two complementary mechanisms work together to implement this protection. The first mechanism is filtering of packets addressed to protected servers. The firewall must screen packets based on source, destination and type of service. One would like to allow only traffic that originates from external clients that have been predetermined by the firewall administrator as being allowed to establish sessions with protected servers. Another policy aspect that one can implement with filtering is to control which servers on the protected side may be accessed by clients on the outside.

The second mechanism is identification and authentication of the individual on the outside establishing the session. Unlike the filtering requirement, a very specific implementation is required for session identification and authentication. The prescribed implementation is that of challenge-response cryptographic authentication using the Digital Signature Standard (DSS) algorithms on the Fortezza card [need reference]. The challenge-response protocol using DSS authenticates the distinguished name of the user attempting to access the server. After successfully authenticating the user the firewall then opens a session to the server the user requests to access.

For purposes of interoperability, a MISSI standard is needed for the challenge-response protocol. This standard is in development. It will require compliance with the NIST FIPS JJJ [1] protocol for challenge-response authentication. It is important to note that this authentication is only for session establishment. There is a concept of continuous authentication using Secure Sockets Layer (SSL) [reference needed] in MISSI. This continuous authentication is client-to-server; the firewall plays no part in SSL.

The initial session authentication provided by the firewall and the continuous session authentication provided by the SSL protocol in the clients and servers complement one another. If the initial authentication at the firewall fails, the session will not be established. If the continuous authentication fails, the session will be terminated by the client or the server. The precise specification for how SSL will be used in MISSI is being developed. Since that development is not yet mature, this paper contains no further elaboration.

4.2 Message-Oriented Protocols

For message-oriented protocols, the MISSI compliance requirements are considerably less than for session-oriented protocols. Packet filtering is the only mechanism that is required for message-oriented protocols. The requirement gives administrators a tool to specify certain external hosts and internal hosts that may communicate across the firewall with message-oriented protocols.

There is no requirement to use DSS for verification of the writer of messages. Likewise, there is no requirement for the firewall to apply DSS to outgoing messages. This is in keeping with the philosophy that the firewall provide complementary services within the overall MISSI architecture. In MISSI, message signatures and message encryption are handled writer-to-reader by the user workstations and the Fortezza card. The firewall needs to provide no additional support for message signatures and encryption.

4.3 Logging and Audit

The firewall is required to audit the session-oriented protocols that traverse from the unprotected side to the protected side. It is not required, or even desirable, to audit all details of the network traffic. The firewall must capture the start of all sessions, all session attempts that are rejected, and all details of authentication failures.

There are no requirements for auditing on message-oriented protocols in the current draft requirements. In an earlier version, there was the requirement that the firewall audit all messages that traverse in either direction. This would have required the firewall to implement a full application for the messaging software to collect all packets comprising a message, determine the identities of the sender and receiver, and record that information along with the time and date. Upon further consideration, the cost to vendors of implementing this option did not justify the benefit to the customer. Hence, the requirement for access control and audit on messages was reduced to packet filtering.

4.4 Assurance

The assurances required for MISSI Compliant firewalls at the SBU level are very basic in comparison to those required for evaluated operating systems at the C2 level. The reasons for such low assurance are two fold. First, most firewalls on the market have no rigorous assurance techniques applied to them. Second, the writers of the MISSI Compliance requirements recognized that firewall technology is evolving so rapidly that the application of rigorous assurances would produce unacceptably long evaluation schedules. The target for MISSI Compliance evaluation and testing is 90 calendar days. With this time constraint a rigorous design analysis is not possible. Therefore the assurances applied are mainly security testing. While security testing is far from high assurance, it is a quantum leap compared to the current environment where many firewalls have had no assurance techniques applied by an independent party.

While it is correct that testing is inconclusive in proving the absence of vulnerabilities, simulation of attacks that the firewall might be subject to in an actual implementation will give useful information about the firewall's resistance to attack. The evaluators will not perform all of the testing, however. The developer is required to have tested the firewall to support assertions that the firewall protects against common network attacks. The evaluators will inspect the vendor's tests, observe the execution of those tests, and perform additional tests at their discretion.

5.0 Interoperability Requirements for MISSI

After satisfaction of security requirements, firewalls must satisfy interoperability requirements for MISSI compliance. These requirements address the challenge-response exchanges for session-oriented protocols, as well as interoperation with other components of the MISSI Architecture. The goal is to have a common challenge-response for session-oriented protocols such as ftp and telnet. Users may have the operational need to access servers at various sites, protected by a variety of firewalls from different vendors. If the firewalls are MISSI compliant, a user's client that implements the MISSI challenge-response scheme should be able to conduct a challenge-response exchange with any of these firewalls. NSA/ISSO has contracted to Trusted Information Systems for development of a prototype challenge-response scheme for ftp and telnet. This scheme makes use of the FIPS JJJ challenge-response standard. [need reference] A MISSI Concept of Operations for Identification and Authentication is currently under development, this will document the MISSI challenge-response scheme for firewall vendors.

For interoperation with other MISSI components, there are requirements for accepting Certificate Revocation Lists and Compromised Certificate Lists. For the immediate future, firewalls merely need to accept these lists as they are manually loaded by the firewall administrator. In the future, MISSI Compliant firewalls will have to be capable of accepting electronic Compromised Certificate Lists from the Certification Authority Workstation. Also for the future, will be a requirement that the firewall have an integrated MISSI Audit Agent that is capable of collecting information about events that occur on the firewall and sending that information to an external MISSI Audit Manager. The rationale for making this a future requirement rather than a current one is that the MISSI audit components are not yet available.

6.0 The MISSI Compliance Evaluation and Testing Program

MISSI Compliance Program vendors will be admitted to the MISSI Compliance Evaluation and Testing Program after they sign an agreement to integrate Fortezza into their product. Before testing may begin, Vendors must show documentation as evidence of their own testing against common network attacks. Products will be prioritized for testing based on a number of factors which may include: the order in which they signed agreements to integrate Fortezza, the market share of the product, customer demand for the product, ability of the product to satisfy the security and interoperability requirements, and government resources available to conduct evaluation and testing.

MISSI Compliance Program for firewalls consists of two parts: Security Evaluation and Interoperability Testing. The entire testing process starting from the vendor providing test documentation is planned to take 90 days.

NSA is currently in the process of determining the applicability of the MISSI Compliance status. There must be a clear definition of what constitutes a significant change that would warrant a new evaluation or test of a firewall. This is a critical issue because of the nature of firewalls and the firewall market. New versions of firewalls are released in quick succession and Firewalls are ported to a variety of platforms. The answer to this question must satisfy security constraints as well as conditions in the Firewall community.

The Security Requirements for MISSI-Compliant Firewalls Protecting Sensitive but Unclassified Environments are being translated into a Common Criteria Protection Profile (CCPP). This CCPP will be used by the government to conduct a low assurance trusted product evaluation. This evaluation will include analysis of administrator and user documentation and possibly examination of security testing done by the vendor. The evaluation team will augment the tests with common attacks the product is likely to face when it is fielded. Due to the large number of Firewalls to be tested and the changeability of the technology and products, NSA is looking into the possibility of using commercial entities to perform testing against these requirements

The second major component of MISSI Compliance is interoperability testing. Interoperability test plans state, at a high level, the functions and features that are tested for each product submitted to the program. Unlike the security testing described in Section 4.4, vendors are not required to have conducted interoperability tests before submission of the firewall to the MISSI

Compliance Program.

Firewalls that are successful in both the security evaluation and interoperability testing components of the program will be given MISSI Compliance Status. This status will be qualified with the version of the requirements applied and the version of the firewall.

All aspects of the MISSI Compliance Program have been developed in coordination with the firewall vendor community. As the requirements and process were written, drafts were distributed to the vendors for comment. By gaining community input, our goal is to achieve consensus on the requirements and process, thereby gaining maximum participation by the firewall vendor community. Development of updated requirements and improvements to the process will be done with full and open exchange with the vendor community.

7.0 On the Horizon for MISSI Compliance

A separate set of requirements are currently being developed for MISSI compliant guards protecting environments processing classified information. The security requirements for guards will require more assurance. Consequently, security evaluations of guards will be more rigorous, and will take more resources and time.

Virtual Private Networking is a capability that many firewall vendors are beginning to include in their products. Users will be able to encrypt traffic from firewall to firewall creating a private network for themselves using the internet. Aside from the obvious data confidentiality benefits VPN reduces the burden of individuals authenticating multiple firewall to firewall sessions or messages. In some cases, additional services can be opened up between the firewalls with the encrypted connection.

As the state-of-the-art in firewall and guard technology evolves, so must MISSI. The MISSI Compliance Requirements and Program will be updated as necessary to include new protection techniques that pervade the market. Hence, products that become MISSI Compliant in 1996, might wish to be tested for compliance with a later version of the requirements when they are updated.

8.0 Summary

MISSI Compliance is the program by which NSA plans to make an impact on the state of internet firewalls and guards. The purpose is to provide the Department of Defense with timely, accurate information on commercial-off-the-shelf firewall and guard products. This will allow services and agencies to make informed decisions on which products to procure and how to use them.

The main thrust is to encourage development of commercial-off-the-shelf products that will fit into the MISSI framework. This includes satisfaction of specific security requirements and interoperability requirements. NSA will conduct evaluation and testing of candidate firewall products to determine compliance with these requirements.

9.0 References

1. United States Department of Commerce, National Institute for Standards and Technology. FIPS PUB JJJ, Public Key Authentication Standard (draft), 1 February 1996.

2. United States Department of Defense, National Security Agency. Security Requirements for MISSI Compliant Firewalls Protecting Sensitive But Unclassified Environments (draft), 9 February 1996.

3. Common Criteria Editorial Board, Common Criteria for Information Technology Security Evaluation, 31 January 1996.

PAPER

on

DESIGNING & OPERATING A MULTILEVEL SECURITY NETWORK USING STANDARD COMMERCIAL PRODUCTS

Abstract: In March 1996, the 2nd Bomb Wing, Barksdale AFB, LA declared initial operational capability on the first multilevel security system (a.k.a. multilevel network or MLN) using only low-cost commercially available products. The MLN integrates the many sources and sensitivities of information (secret and unclassified) necessary for a commander to effectively command and control global bombing operations. We developed and implemented the MLN for two reasons:

- First, to reduce the number of terminals each command and control center (C^2) operator must use. Multiple non-integrated systems and the technical necessity of separating classified and unclassified systems have created enormous system overhead and operator training inefficiencies - base and Air Force wide. In many operational areas, real estate is at a premium and reducing required floor or table space would also improve the work environment. Reducing the number of garrison terminals needed could eventually affect deployed operations, where less combat support weight means more combat weight could be transported.

- Second, to reduce operational costs. Costs are reduced by buying commercial products. Savings are enhanced by the commonality of parts among various operational systems as they connect to the network. Training costs will decrease as new operational systems are added to the network because a common human-computer interface would exist between systems.

The MLN is working and the single most expensive item is the operating system at roughly \$3,000 each (\$1900 each with a site license). The MLN is already a model for other C^2 centers and continuous refinement will only improve its desirability.

Prepared by:

Richard A. Griffith & Mac E. McGregor
C4 Techology Validation Office
DSN 781-3777 Commercial (318) 456-3777
Fax: DSN 781-2638 Commercial (318) 456-2638
E-mail: mcgregor@c4tvo.barksdale.af.mil

Prepared for:

AIR FORCE C4 TECHNOLOGY VALIDATION OFFICE
245 Davis Ave. East, Suite 2
Barksdale AFB, Louisiana 71110
Contract Number: BA95218AFO Project Element Plan 04

U.S. Air Force Publication Release Authority

JILL N. ALTMAN, Lt Col., USAF
Director, C4 Technology Validation Office

DESIGNING & OPERATING A MULTILEVEL SECURITY NETWORK USING STANDARD COMMERCIAL PRODUCTS

ABSTRACT

In January 1996, the Air Force declared initial operational capability on its first multilevel security system (a.k.a. multilevel network or MLN) using only low-cost commercially available products. The MLN integrates the many sources and sensitivities of information (secret and unclassified) necessary for a commander to effectively command and control global bombing operations. We developed and implemented the MLN for two reasons:

- First, to reduce the number of terminals each command and control center (C^2) operator must use. Multiple non-integrated systems and the technical necessity of separating classified and unclassified systems have created enormous system overhead and operator training inefficiencies - base and Air Force wide. In many operational areas, real estate is at a premium and reducing required floor or table space would also improve the work environment. Reducing the number of garrison terminals needed could eventually affect deployed operations, where less combat support weight means more combat weight could be transported.

- Second, to reduce operational costs. Costs are reduced by buying commercial products. Savings are enhanced by the commonality of parts among various operational systems as they connect to the network. Training costs will decrease as new operational systems were added to the network because a common human-computer interface would exist between systems.

The MLN is working and the single most expensive item is the operating system at roughly $3,000 each ($1900 each with a site license). The MLN is already a model for other C^2 centers and continuous refinement will only improve its desirability.

KEYWORDS

Air Force B1 B2 C^2 CMW Command and Control Compartmented Mode Workstation
MLN MLS Multilevel Network Multilevel Security System SCO SecureWare UNIX

PLAYERS

The Second Bomb Wing is the host organization at Barksdale AFB, LA. This fully combat operational B-52 wing can bomb any point on earth and return without landing at another base. This capability was proven when B-52's departing Barksdale, bombed Iraq during Desert Storm and returned to Barksdale. The nerve (C^2) center for such an undertaking is the command post. All information necessary for force deployment feeds into the command post by telephone, radio, and a myriad of computer systems and networks. From the command post, the commander develops, organizes, and executes the battle plan.

The Command, Control, Communications, and Computers Technology Validation Office (C4TVO), operating location B of the Air Force Communications Agency (AFCA) at Scott AFB, IL, is also at Barksdale. The C4TVO's charge is validating the latest commercially available technologies and integrating them into the operational Air Force. The purpose of this mission is to enhance combat operations by applying technology:

a.) Without the long research and development lead times required by designing systems from scratch,

b.) Using commercial specifications instead of the more specialized military ones,

c.) To act as a force multiplier through reduced combat support payloads, reduced personnel requirements, system simplification, or reduced operational cost.

This proximity to an operational unit permits the C4TVO to evaluate new concepts and technology at the tip of the spear instead of in laboratories separated by distance and occasionally the reality of operational needs. The location also permits rapid project changes or redirection, including cancellation, without losing huge investments in time or sunk development costs.

PROBLEM IDENTIFIED

The command post has sixteen major computer application systems that are or will be connected to it. These systems were all designed for their separate purposes before compatibility across major systems was a concern in system development. Inside the command post, there are mission planners, aircraft maintenance controllers, and others whose system access requirements are different. In addition to the numbers of systems to which each person needs access (anywhere from 1 to 16), each person may require access to only a certain classification level (secret or unclassified) of a given system. Without a course change, command post members would require unnecessary movement about the command post to access various systems as battle stations became heavily populated with incongruent terminals. Hence, the "fog" in the fog of war would thicken. The command post needed a better way of doing business.

PROPOSED SOLUTION

Beginning in November 1993, wing operations, AFCA, and C4TVO representatives developed a B1 assurance level MLN (having B2 operational features) with two main objectives:

- First, to reduce the number of terminals each C^2 operator must use. Multiple non-integrated systems and the technical necessity of separating classified and unclassified systems have created enormous system overhead and operator training inefficiencies - base and Air Force wide. In many operational areas, real estate is at a premium and reducing required floor or table space would also improve the work environment. Reducing the number of garrison terminals needed could eventually affect deployed operations, where less combat support weight means more combat weight could be transported.

- Second, to reduce operational costs. Costs are reduced by buying commercial products. Savings are enhanced by the commonality of parts among various operational systems as they connect to the network. Training costs will decrease as new operational systems were added to the network because a common human-computer interface would exist between systems.

As the system design progressed, it became apparent a successfully operating system would have applicability in all active and reserve Air Force command posts. Although not a major objective of the Second Bomb Wing host, portability to other command posts was always considered and design simplicity was the means to portability.

NETWORK DESCRIPTION

The MLN accesses unclassified and secret information from a single terminal type known as a compartmented mode workstation. Data confidentiality, integrity, and availability are maintained by combining a workstations' trusted computing base with technical and traditional procedural security measures. The network has unclassified and secret gateways and routers. Each workstation labels data unclassified or secret and transmits information to the proper gateway and router. Each gateway has an internal unlabeled and multilevel network interface card. The routers act as a firewall; hiding the network from the outside world. Network security is increased by prohibiting all common UNIX file transfer services since there are no operational requirements for them. All communication (e.g., electronic mail) beyond the firewall will be to mail hosts where aliasing will further protect the network by hiding MLN addresses from the outside. MLN users will have to pull their mail from the mail host rather than have it pushed to them. All MLN users are cleared for secret although they will not all have need-to-know access to all information within the network. Therefore, the security mode of operation is system high. Identification and authentication within the MLN is through user identification and passwording.

The security testing and evaluation team's methodology was to match the vendor-advertised security features against those of the MLN security policy. Where the advertised features met the security policy the team attempted to prove it or disprove the advertised feature. For those features not meeting the policy, we worked with the vendor to eliminate or mitigate the weakness. The penetration team assumed the positions of unauthorized users outside the MLN and authorized MLN users with bad intentions. They tried to penetrate the system configured in two ways - one as we intended the MLN to operate and the other with full customary UNIX file services available. This was to document, for potential follow-on MLN users, the disadvantages associated with full UNIX capabilities.

The MLN will be fully operational in the command post before any expansion beyond the command post's boundaries. The initial classified system connecting to the MLN is the Wing Command and Control System (WCCS). WCCS provides decision making information like weather, logistics, aircraft mission capability, etc. to the battlestaff for exercises, crises, and war. The initial unclassified system connecting to the MLN is the Core Automated Maintenance System (CAMS). CAMS provides the commander the maintenance status of all operational assets. There are no specially designed hardware or software items in the network. The most expensive item is the SecureWare, Inc. CMW+ 3.0 operating system - a secure version of SCO UNIX. The license price is about $3,000 each for ten licenses. A new site licensing agreement with SecureWare will bring this cost to around $1900 each.

DESIGN AND OPERATIONAL ISSUES AND LESSONS

An operational concern in designing the MLN was classified and unclassified data aggregation. If intruders were to compromise a fully operational (with all sixteen mission applications) MLN, they could presumably compose the full air order of battle. This knowledge, and knowing the unclassified networks beyond the firewalls had internet access, made the Designated Approving Official decide various MLN components would effectively meet the B1 and B2 assurance levels. Those components beginning at the MLN's secret gateway would have the mandatory access control feature of labeled security (B1). Those components between the MLN's secret gateway and the unclassified gateway would have the added assurance of a trusted path, least privilege, and proof the system can't be spoofed (B2).

The early challenges occurred when the OS vendor, switched from a previously tested and security certified OS version 2.3 to the current version 3.0. During our security testing and evaluation process, we discovered several security-related problems which required considerable coordination with SecureWare to resolve. Such problems are normal in any software design and development process. The vendor completed and delivered the patches. The patches passed the subsequent security and penetration testing and are now operational.

Other issues will arise as we add more and varied applications to the MLN. The main one with the first application suite, WCCS, were caused by differing system architectures. For example, the MLN was designed to use low-cost commercially available products like Wintel 486 systems. Initial MLN performance in such areas as screen refresh rate, etc., was slower than on WCCS terminals. This existed because MLN terminals are software driven and they were competing against WCCS diskless workstations where the X Window software was on a RISC chip. Upgrading MLN terminals to 90MHz Pentium processors seems to be the near term solution in our early trials. Faster processors, as they become readily available, will be the longer term solution.

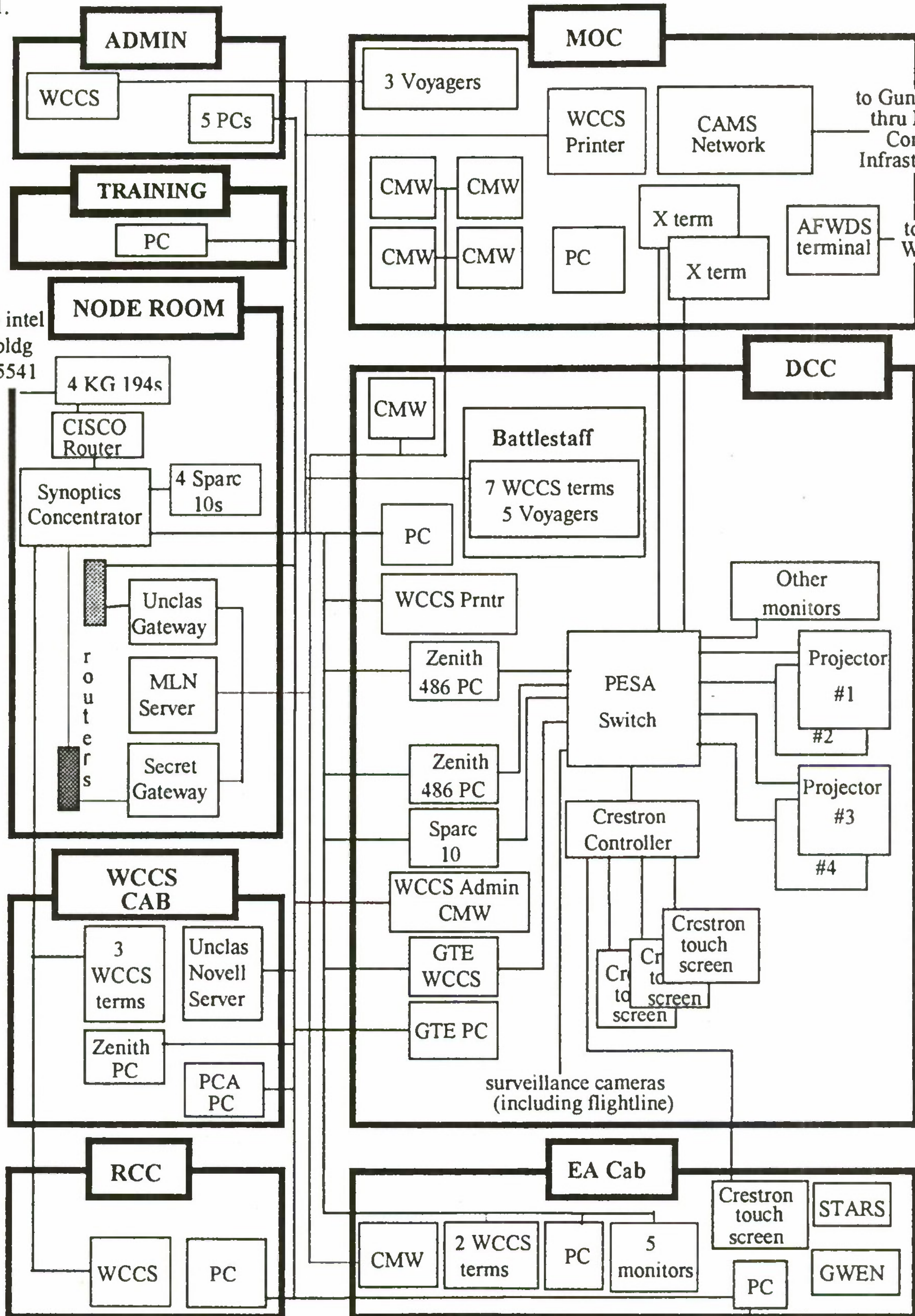
An external incident directly affecting the MLN resulted from new WCCS OS versions being released with different software configurations which adversely affected the MLN interface. The new releases would not run, or would cause the MLN to crash. Our coordination with the WCCS program office (who is not specifically tasked with considering MLN requirements in their own system design) earlier in their design and release cycle would prevent this problem. These type problems will lessen as the MLN becomes a standard.

The final problem encountered to-date is a software licensing one, which SecureWare is changing. Our original SecureMail license permits seven users on the system, as we requested. We had only seven user terminals and that licensing arrangement appeared to meet our requirements. After receiving the software, we learned the software would only accept seven users in the system. What we truly needed was an unlimited number of users with a limit of any seven simultaneous users. Better communication between ourselves and the vendor could have eliminated the delay in becoming fully operational until the newly licensed software package arrives.

SYSTEM OVERVIEW

Multilevel Network Logical Configuration

Diagram 1.



Colored lines are classified   Red lines are MLN additions   to bldg 5546 concentrator

MULTILEVEL NETWORK CONFIGURATION KEY

AFWDS	Air Force Weather Data System
CAMS	Core Automated Maintenance System
CISCO	Private company name
CMW	Compartmented Mode Workstation
Crestron	Private company name
DCC	Display Control Center
EA Cab	Emergency Actions cab
GTE	Private company name (formerly Gray Telephone and Electric Company)
GWEN	Ground Wave Emergency Network
multi-level network	Multi-level Network (MLN)
MOC	Maintenance Operations Center
PCA	Private company name
PESA	Private company name
RCC	Reports Control Center
STARS	Strategic Arms Reduction System
Synoptics	Private company name
Voyagers	Sun corporation portable computers
WCCS	Wing Command and Control System terminal
X Term	NCD corporation dumb terminals running WCCS with an X Window user screen
Zenith	Private company name

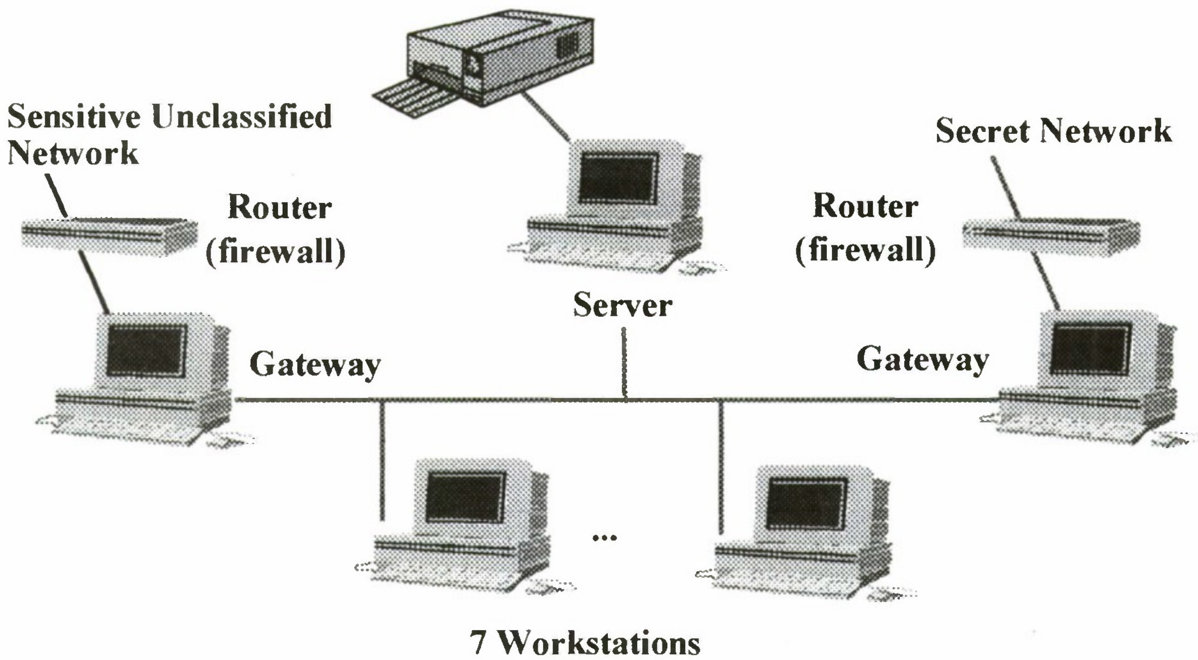
MULTILEVEL NETWORK COMPONENT CONFIGURATION

Hardware Identification. Table 1 shows the hardware on each workstation. Diagram 2 shows the hardware configuration.

Table 1. Hardware list.

Quantity	Item	Description
1	server	i80486 CPU, 2GB & 1GB hard drives, 3.5" floppy, 1 DAT 4mm tape drive
1	secret gateway	i80486 CPU, 1 hard drive
1	unclassified gateway	i80486 CPU, 1 hard drive
7	user terminal	i80486 CPU, 1 hard drive
1	laser printer	HP Laserjet 4 w/2MB memory

Diagram 2. Hardware Configuration



Red lines carry classified data

Table 2. Assurance Levels

Assurance Level	MLN Component	Boundaries
C2	Unclassified Segment	Begins at the unclassified gateway and includes the unclassified router
B1	Security Services	Begins at the secret gateway and includes the secret router
B2	MLN Segment	Includes the printer, user terminals, server, and all connecting lines

Hardware Notes.

All hosts are different Intel x486 platforms with 15" color monitors (to be upgraded to 17"). The disk drives range from 350 MB - 1 GB and the memory ranges from 28 - 32 MB.

Four compartmented mode workstations will be in the command post's Maintenance Operations Center (MOC), one in the Emergency Actions Cabinet (EA Cab), one in the Battlestaff area, and one in the Data Control Center

(DCC) for system administration. The server and gateways will be in the node room behind the EA Cab, each placed 1 meter apart.

Each gateway has two ethernet boards. They and the administration machine may only be accessed at the console. The gateways do no packet filtering. However, the operational user is considering using tcp_wrappers. Hot spares are planned for the gateways; two at the unclassified sensitive and two at the secret interface. The gateways will be statically routed.

The server will have at least two 2GB disk drives and at least 64MB memory.

There will be one multilevel printer (an HP 4) on the MOC floor. All output will be labeled at the appropriate classification level. Output for applications that reside outside the MLN (e.g., WCCS) will go to the normal application printers. For example, output for WCCS will print on the WCCS printer just as output from CAMS will print on a printer on the unclassified segment.

There are no modems. If dial-up service is required, then STU-IIIs will be used.

The floppy drive will not be accessible by the typical users. There may be a few users with floppy drive access. Drives df0 and df1 have been disabled and the drives cannot be accessed as a: and b: drives unless the user has the dosfloppy privilege.

Software Identification.

Table 3. Application Software

Product Name	Version	Vendor	Functionality
Office Professional		Microsoft	Spreadsheet, Slide Preparation, Word Processing, Database Management
SecureMail	2.0	SecureWare	Electronic Mail

Software Notes.

The compartmented mode workstation operating system is SecureWare 3.0 (CMW+ for SCO Open Desktop).

The compartmented mode workstation window system is an X-window environment.

The compartmented mode workstation includes MaxSix software, version 2.0, which provides additional network-related security capabilities. MaxSix provides the mechanism establishing authorized connections to high- and low-side systems from the appropriately labeled window through the correct network interface. It also labels the incoming data according to the assigned sensitivity label of the network interface.

Two Trusted Network (TNET) databases are used by MaxSix to implement security policy. They are the TNET Interface Database (TNETIDB) and the TNET Remote Host Database (TNETRHDB). The TNETIDB file specifies the default security attributes of datagrams associated with each network interface (each ethernet board). The TNETRHDB file specifies the security attributes associated with hosts residing on a network. For example, TNETRHDB specifies whether a host is another TNET host (e.g., another compartmented mode workstation) or a non-labeling host (e.g., a generic UNIX system). Also, TNETRHDB specifies the security accreditation range for the host. The host accreditation range is a set of minimum and maximum sensitivity labels representing those sensitivity levels that can be processed by the host as a whole. Table 2 shows the application packages installed on each workstation.

SUMMARY

The network is currently undergoing operational validation with a multilevel electronic mail system, one of the sixteen applications operating at the secret level (WCCS) and one operating at the unclassified level (CAMS). The MLN appears to be meeting the two design goals. However, until the MLN operational evaluation is complete, this should be considered an early, but reasonable conjecture.

Note: ***Extra Drivel Not used in the above Paper***

One of the MLN's advantages is its ability to be a multifunctional user terminal. Besides tying in to several application specific networks, it also contains Microsoft's Official Professional Suite. The only application causing the MLN any problems was the Excel spreadsheet. We initially configured the MLN to allocate 4MB of RAM to Excel. The operational test showed it wouldn't work until the allocation was changed to 12MB.

REAL WORLD ANTI-VIRUS PRODUCT REVIEWS AND EVALUATIONS – THE CURRENT STATE OF AFFAIRS

Authors' note: The original work upon which this paper is based discussed problems and alternatives relating to the evaluation of anti-virus software. It was published with the hope that users and developers would provide us with suggestions for developing evaluation methodologies which would work in the real world. Our goal was to help create viable evaluation criteria which corporate security managers could apply when selecting an anti-virus product. Since the original publication of this paper, we have received suggestions from many anti-virus product vendors, security personnel, magazine evaluators and reviewers and government representatives. This revision reflects the new direction anti-virus product certification appears to be taking in the "real world" today.

Sarah Gordon (sgordon@dockmaster.ncsc.mil)

Richard Ford (rford@commandcom.com)

Abstract:

This paper will discuss frequently encountered errors in the evaluation process relative to anti-virus software selection by examining some of the methods commonly used by corporate and governmental personnel working in the area of Management Information Systems (MIS). In addition to discussing inherent problems, we will suggest alternative methodologies for evaluation. We will examine commercial certification processes, as well as the Information Technology Security Evaluation and Certification (ITSEC) approach, as possible models for anti-virus product evaluation and certification. Finally, we will discuss ways in which the information which is currently available may be used to help select anti-virus software which is both functional and cost efficient.

Introduction

The evaluation of anti-virus software is not adequately covered by any existing criteria based on formal methods. The process, therefore, has been carried out by various personnel using a variety of tools and methods. Some of these tools and methods should be part of the evaluation process; others can provide misleading or damaging information resulting in increased exposure to computer viruses. Areas of the evaluation which are relatively straightforward include the elimination of products which are unsuitable for your environment, the cost of the software, comparison of vendor pricing policies and licensing conditions and assessing compatibility requirements. In all of these areas, you must of course anticipate future growth; for instance, if you are planning to add platforms or anticipate many users taking work home, you will need to rule out software which does not support multiple platforms or which does not allow for acceptable home use pricing packages. Products must of course be well documented and easily configurable. Transparent operation is required, as products requiring large overhead tend to invoke removal or circumvention on the part of the user or administrator. These areas of examination are important; however, there are other aspects of the selection process which are even more critical. You may even depend on evaluations you don't know anything about, as in the first two cases we will examine. Unfortunately, as we will show, there are serious problems with *all* of the evaluations on which people are currently relying.

"It is unfortunate, but a large majority (say 90 percent) of the current anti-virus tests published within the last couple of years are worthless, or even worse than that, purposefully made misleading." [1].

We will examine this claim, beginning with the types of evaluations you may find yourself having to base your decision upon. The following, based on "Real-World Anti-Virus Product Reviews and Evaluation" [2], illustrates that the majority of methods are impractical.

The Provider of Friendly Advice

Managers seriously underestimate the power of "the friendly recommendation" by friends, or colleagues who have "used xyz anti-virus and it worked just great". However, with the limited time and resources many companies have to investigate what constitutes a viable anti-virus solution, the influence of the friend should be duly noted. The inherent problems in relying on the recommendation of friends, even knowledgeable friends, result from both the competence level of the friend and the variance in needs of users. For instance, if the main requirement for "the friend" is that the system provide for a means of circumventing a scan, whereas your need requires non-circumvention, you would be ill-advised to select a package which allowed for easy circumvention. Variables such as packaging, pricing, and speed are all subject to interpretation, and the interpretation will be greatly influenced by the needs of the individual who does the reading.

A much more serious issue is related to claims of performance in the area of actual virus detection. Consider the claim of a friend that "the program worked fine. My system is virus free!". The question here is "How does he know he never had a virus?" If he is using a product which misses viruses, he may think he never has had one when in fact he has. He may also be relying on what he has heard from a friend of a friend, who really likes anti-virus because it is the one he is familiar with. People are very much influenced by name-recognition. However, do you want to trust the security of your data to a product based on its name? We argue that you should base your decision on the actual performance of the product. Unless 'the friend' happens to be particularly skilled in anti-virus product evaluation metrics and methodologies, it is probably not a good idea to trust his or her advice.

The Employee (or the employee's friend, colleague or Internet acquaintance)

The Employee resembles "The Provider of Friendly Information" in many ways, with the additional attribute of feeling somewhat responsible. Employees become "virus experts" by reading virus information message areas on various on-line services. They may obtain some viruses to "test" the efficacy of software you have, or are considering purchasing.

You need to be concerned about the employee *not* because he/she is acting out of any form of malice; on the contrary many employees feel they are helping you by becoming "experts" in virus testing. However, a thorough understanding of product evaluation is not something an employee can learn in their off hours by "beta-testing" some anti-virus software and recommending it to people because "it caught a lot of viruses".

The reasons such well-meaning expertise is ineffective relate in part to the technical skills required to construct and perform a meaningful test. Can the employee disassemble and replicate samples to ensure the test-set is clean; i.e. that test samples are actually viruses and not corrupted files? Is the employee capable of judging the efficacy of the removal and terminate and stay resident (TSR) modules of packages? What tools does he have at his disposal? Does the employee have a dedicated test machine upon which to perform tests and has he or she studied the subject enough to do the job correctly for you? It is unlikely that most companies have the resources to answer 'yes' to these questions, yet we see company virus representatives talking about their in-house evaluation of products. We suggest that their evaluation is not only inadequate, but it can also be harmful to the integrity of the company data.

The employee who has been granted some official status may be familiar to you as one whom you have designated to do in-house evaluation - a member of the technical support team or a programmer. However, even a technically competent employee is not likely to be able to carry out tests of the quality which you require in order to evaluate a product fully. You must remember that "technically competent" in programming or network administration does not imply "technically competent" with computer viruses.

The Computer Magazine (non-virus/security specific)

The Computer Magazine evaluator /reviewer is in a unique position; he holds a lot of influence over the public, while at the same time usually having insufficient experience in the field to provide accurate information. This frequently leads to reviews which rely on incorrect assumptions. As an example, a well-known computer magazine recently hosted an on-line forum during which the magazine "expert" stated certain boot sector viruses can infect the fixed disk of an otherwise clean machine simply by the user typing the command "DIR" with an infected diskette in the A: drive. Apart from a lack of technical ability and information, a computer magazine is unlikely to have a large and clean collection of computer viruses. Therefore, the reviewer is likely to take one of the following approaches:

- Carry out a test on a very small "collection" of "viruses", gathered from friends or colleagues.
- Approach an anti-virus software developer for a collection of "viruses".
- Obtain a collection of "viruses" from a virus exchange bulletin board system (vX BBS), ftp site, the World Wide Web, or a publicly-available virus collection such as those available on CD-ROM.
- Use a virus "simulator" to test the detection capabilities of products.

Unfortunately, any tests based on "samples" obtained in this manner lead to questionable results. We shall examine the problems with each approach in turn.

Using a small collection of viruses is clearly an unacceptable way to carry out a product evaluation. In order to test a product's detection capability, tests should be carried out against at the very least all those viruses known to be in the wild (ITW). We suggest "The Wildlist", by Joe Wells as a good starting criteria for detection. Testing against only a few viruses will not give an accurate impression of a product's ability to meet the real threat. However, such tests have been done and the results printed. We are even aware of one review which based its final detection results on a test-set of only 11 viruses [3].

The problem with using a vendor's virus collection is equally obvious: bias. A vendor could simply doctor the test-set so that its own product would score well, or release test-sets which will show the product gradually improving with time.

There is, of course, the additional concern of magazine reporters' and journalists' technical competency in not only replication and analysis but in management of virus libraries. It is important to make sure the viruses used for testing are not only real, but that they do not inadvertently escape and cause harm to unsuspecting users, or result in liability to the magazine. We know of several cases where computer viruses were inadvertently released on computer diskettes distributed with computer magazines (although we are *not* aware of any link between this and the testing and reviewing of anti-virus products).

The issues raised by obtaining a virus collection from a vX BBS or the Internet are more subtle. In these cases, the reviewer has no way of ensuring that each sample is actually infected by a virus. Virus collections obtained in this way are frequently badly organized, containing a large number of corrupted or uninfected files. Detection tests carried out against such a collection are not likely to be accurate, and will discriminate against the better products. This is summed up by Tanner [4] in "A Reader's Guide to Reviews", which looks at some of the ways to fix a test made on two fictitious products, Grottyscan and Wonderscan:

> You'll need a test suite. Ideally, you should get it from Grotty Inc. You might find that Grotty Inc. don't have a virus library, in which case, you should find a collection of files that contain viruses and also lots of corrupted and innocent files. That way, if half the files you use are not viruses, the GrottyScan score of 30% doesn't look too bad compared with the 40% that the best product got.

The article continues onwards in a similar vein, and highlights several of the other ways to bias a test, either intentionally or inadvertently.

In the case of a fixed collection, like that available on CD-ROM, there is yet another issue: anti-virus product developers have had unrestricted access to the actual samples against which the test will be carried out. This is a problem because if the scanner manufacturer has access to the test collection, it is a trivial exercise to alter the product so that every sample in the test-set is deemed to be infected, regardless of its state. Although the scanner may detect the samples of the virus stored on the CD-ROM, it may be unable to detect further replications of each sample. This is particularly true in the case of polymorphic viruses, where test results are invalidated if the software developer has copies of the actual samples used during the detection tests. Thus, using a fixed collection of viruses to which every vendor has had access provides little real information about real world scanner performance.

We have observed the development of a disturbing trend: testers using virus simulators to test products. This is unacceptable for several reasons. One of the more popular simulators creates .COM and .EXE files, and provides supplemental Mutation Engine (MtE) samples. The .COM and .EXE files simply print a message to the screen and exit. It is clearly unacceptable for an anti-virus product to detect such activity as viral. Although these files also contain virus signatures (non-functional "fragments" of virus code), anti-virus technology has by necessity evolved in such a manner as to render detection of such simulated "viruses" a useless measure of the product's actual capability. According to a report published by Luca Sambucci, of the Italian Computer Virus Research Institute, tests using simulated viruses are "misleading and in some cases harmful".

In comparative tests we conducted using both simulated viruses and real viruses, we found that while the scanners we tested detected all of the real viruses, only one scanner detected any of the simulated viruses. Tests performed on simulated (fake) viruses do not necessarily accurately reflect the detection capabilities of a product [5]. [Note: The EICAR test file, developed by the European Institute for Computer Anti-Virus Research, should not be considered a simulated virus; rather, it is a program which scanner developers have deliberately chosen to detect. While it is not useful for measuring the comparative detection ratio of products, it may be used to test installation of anti-virus products. It is available from most vendors as well as from http://www.commandcom.com/html/eicar.html.]

The use of simulated polymorphic viruses presents yet another problem. In the most widespread virus simulator available, the Dosen Rorenthal Virus Simulator (this and other simulators are discussed more completely in [5]), the polymorphic viruses supplied *are* viruses, but have extremely limited propagation, infecting only certain designated goat files. Since these "viruses" cannot infect any other executables, the ability of a product to detect them is meaningless in terms of actual protection for the user; a vendor may of course decide to detect them for purely commercial or academic reasons. One possible risk is that these "test viruses" can be modified to be malicious in their action. Thus, many products detect these files "just in case". Such test viruses provide fodder for test libraries, but little else. The creation of computer viruses for any "testing" purpose is both unnecessary and unethical, and the International Federation for Information Processing (IFIP) has issued strong positional statements against such creation.

Assuming that the magazine has managed to gather a number of real viruses without obtaining them from a vendor, a CD-ROM, simulator or unverified source, magazine evaluations rarely test anything other than user interface, configuration issues, and the detection rate of the non-resident scanner. While these factors are important, in no way do they comprise a comprehensive evaluation. Yet, many MIS managers base their choice of anti-virus software on

"Editor's Choice" Awards, or magazine reviews. Such awards are a valuable measure of some aspects of performance, but can be subjective and should not be considered in any way a complete product evaluation.

The Computer Security/Virus Magazine

Reviews published by computer security/virus specialist magazines can provide you with information which may be useful in determining a product's strengths and weaknesses because they have a distinct advantage: the reviewers generally have both experience and a specialized knowledge of anti-virus products. These reviews tend to be well done and informative, focusing on the ability of products to meet published criteria.

Many reviews published in this type of journal attempt to focus on the threat posed in the real world, concentrating on those viruses which are known to be ITW. *Virus Bulletin*, for example, uses the Wildlist to form the "In The Wild" test-set for file viruses. This examination of the real threat, frequently coupled with tests which take into account the product's performance against a number of different infection strategies leads to in-depth reviews of a good quality. Unlike most magazine reviews, the specialist magazines are almost guaranteed to carry out tests against real viruses, and are a source of accurate detection results. Unfortunately, even these reviews have their share of problems. For instance, although having now instituted a totally ITW Polymorphic test suite, *Virus Bulletin* tests on boot sector viruses and polymorphic viruses have in the past included viruses which are not in the wild, leading to some confusion in interpretation of test results. *Secure Computing* published in their May 1996 Lead Review, tests which measured the ability of a program to detect its "Advanced Polymorphic" test suite. The scanners were tested on a collection of polymorphic viruses which were damaged in some way and would not either replicate or execute. Samples which do not replicate are of course not viruses, and while the tests were correctly interpreted, they are also a completely meaningless measure of actual protection.

Another commonly cited problem is that of tester independence. The two most well-known magazines which regularly test anti-virus software (*Virus Bulletin* and *Secure Computing*) have both been associated with producers of anti-virus products: *Virus Bulletin* with Sophos (Sweep) and *Secure Computing* with S&S International (Dr. Solomon's Anti-Virus Toolkit). While there is little evidence of deliberate bias in the review methodology and choice of test-set, these links are worth considering, and are frequently cited by disgruntled product manufacturers. How much bias there is in reviews carried out by such journals is impossible to quantify, but we stress that assuming bias when there is none is just as damaging as not being aware of bias when it is present.

Another problem is the limited nature of the tests. Non-resident scanners are the most commonly tested modules of anti-virus software. The "best" product for a company must be able to operate in a variety of environments, and under several different conditions. Most reviews (particularly comparative reviews) are in reality only measuring one aspect of product performance. Properties which are trivial to measure, such as the rate of false-positives, are often overlooked, and disinfection or detection in memory is rarely if ever tested. Due to time constraints and cost, however, it is not practical for even a specialist magazine to test all aspects of product performance. *Virus Bulletin* has taken some positive steps in this area, however, and is in the process of adding memory-detection and disinfection testing.

Finally, the information given in these magazines is often highly technical in its nature, and it is easy for the reader to suffer from an information glut, obscuring the true strengths and weaknesses of the product. An example of this is the *Virus Bulletin* comparative review of virus disinfection software [6], where the results detailed which parts of the EXE file header had been altered - data which most users would not know how to interpret.

Even with these problems, the virus and security specific publications offer possibly the best analysis of the detection capabilities of anti-virus products.

The Independent Professional Evaluator (IPE)

There are some independent reviewers who posses the expertise to conduct a meaningful review. One good example of such a reviewer is Rob Slade, a frequent contributor to Virus-L and the Fidonet Virus echo and author of several books on computer viruses. His reviews illustrate a major difficulty experienced by others who are attempting to carry out reviews: lack of resources. However, in Slade's case much of this is made up for by his experience and expertise. While Slade represents all that is best about the IPE, there are many self-appointed experts who have neither his experience nor expertise. There is no easy way to discriminate between those who are qualified to carry out such a review and those who are not. One only has to recall the glut of virus "gurus" who appeared during the "Great Michelangelo Scare" to see the problems which you will have deciding how much reliance to place in independent reviews of software.

Another notable reviewer (and founder of the Italian Computer Anti-Virus Research Institute), Luca Sambucci, has provided independent testing to computer magazines since 1992. His anti-virus tests are thorough and competent; however, he has not released a result for almost one year. He still conducts tests, and is primarily concerned with scanner-based detection. He includes explanations of test terms in his test documentation, and gives developers the opportunity to comment on the tests -- as part of the actual test document. Although Sambucci's tests are good, it is difficult to pick his results out from those of the other self-appointed experts without considerable expert knowledge.

The signal-to-noise ratio surrounding the IPE can be observed by monitoring the electronic traffic which accompanies reviews by other independents. Generally the complaints revolve around the lack of performance by a specific product and the qualifications (or lack of them) of the IPE. The publication of qualifications of testers is an important aspect of a complete evaluation and is critical in the area of product certification. The need for this is built in to the very fabric of the Trusted Computer Security Evaluation Criteria (TCSEC): 'Certification should be done by personnel who are technically competent to assess the system's ability to meet the security requirements according to an acceptable methodology' [8]. Thus, without an in-depth knowledge of the IPE's qualifications and history, you should assign little (if any) weight to his results.

The Commercial Evaluator

Probably the most well-known commercial evaluators in the USA are Patricia Hoffman (*VSUM*) and the National Computer Security Association (*NCSA*). Currently there are serious problems with both of these evaluation services, although since the earlier study we have observed some of these problems have been addressed. In particular, *NCSA* has made significant revisions to its test methodology and criteria. The following list of problems, therefore, will be followed by a notation of the changes adopted by *NCSA*.

In both cases, "certification" is not in fact a thorough testing of the entire product, but a test of the scanning engine, carried out by running the product on a large collection of files which the evaluator claims are infected. In other words, the only property of the product to be evaluated is the non-resident virus scanner's ability to detect viruses. No tests are made on other critical areas of the product, particularly, the real-time protection offered or virus disinfection.

An epidemiological overview of viruses shows that although there are over 8000 viruses known for the IBM PC or compatible, there are less than 300 ITW (that is, actively spreading on PCs). A list of such viruses is maintained by Joe Wells. By collating statistics provided by over 30 contributors from many different countries, Wells tracks those viruses which are spreading. Participants in the list include all the major anti-virus software developers, and several independent researchers. The list is broken down into two parts: an upper list, for viruses which have been seen by two or more participants, and a lower list, which is made up of those viruses seen by only one participant.

Analysis of Wells' list shows that the real threat to computers is posed by less than 300 different viruses; if a computer were protected with a scanner which detected just these viruses, well over 99% of the total threat would be covered [9]. Thus, any intelligent test of anti-virus software must weight the detection of these wild viruses *significantly* higher than detection of other non-wild (Zoo) viruses. In essence, tests of Zoo viruses such as those performed by *VSUM* and *NCSA* provide almost no information on the suitability of a virus scanner for a real-world application.

Such tests, within certain limits, do give the reader quantitative information. However, they are highly limited in their applicability to anything approaching formal certification. Certifications like this fail to provide a fully functional baseline for several reasons; foremost among them the only information given is the overall detection rate of the scanner. No information is given about how well the product performs against the threat which users face in the typical office environment. In an extreme case, it would be possible for a product which could not detect any virus which is in the wild to still be certified. [One test which it is valuable to apply to any evaluation of anti-virus software is to examine how a simple batch file which identified every file it was presented with as infected would fare using the test methodology. Under any test which just measures overall infected file identification, such a batch file would get the highest possible score - a result which is obviously misleading.]

The tests by these commercial evaluation/certification services also do not take into account products which have "review" modes, although this problem is in the process of being reviewed by the Anti-Virus Product Developers (AVPD) Technical Committee, a vendor organization composed of technical representatives of member companies. The problem of review modes is a thorny one to solve. Consider a product which changes the way in which it operates when it detects more than a certain number of viruses on any one scan, loosening the criteria which it uses to identify files as infected. Such a scanner would do well on a test carried out against a large number of infected files. However, its detection rate in the test would not reflect its detection rate against the real threat, as usually one would be relying on the scanner to scan incoming diskettes, when the product would apply its stricter criteria for detection.

Finally, there is the question of who has access to the test-set. If software developers are allowed unrestricted access to the actual samples used for the certification, an unscrupulous vendor could change its scanner so that it identified every file in the test-set simply by carrying out a search for a hexadecimal scan string. As the vendor's only interest is finding files in the test-set, the search pattern would not even necessarily be taken from the virus: it would just need to be something capable of identifying that particular file. In the case of polymorphic viruses, this would result in the scanner detecting the samples in the test-set, but no other replications of the same virus. However, denying the developer *any* access to the test-set raises questions about the quality of the test-set: are the files in it actually infected? How much can the test results be relied upon if there is no peer review of the test samples? [3, 7]

In 1995 the *NCSA* certification scheme [then under the direction of one of the authors, RF.] was altered to reflect new, more stringent criteria. A 100 percent detection rate of ITW viruses using the Wildlist as the criteria for such viruses was implemented, with a two month lag time in testing to allow vendors sufficient time to implement detection, taking into account Beta test and shipping cycles. Developers were disallowed access to any samples used in actual testing in the Wildlist portion of the tests. Developers who were members of the AVPD were given access to *replicants* of samples should their product fail to detect them during a certification test. This has the dual benefits of ensuring that the samples are actually fully functional viruses while disallowing the possibility of the developer implementing detection for the file rather than the virus. As a commercial certification, the PC version of the *NCSA* scheme found acceptance as a minimal criterion by which users could judge effective detection rates of scanner portions of anti-virus software. According to *NCSA* Spokesperson Pam Martin, "Any certifications performed by *NCSA* are performed strictly for end users. There is no attempt to mimic or supplant the ITSEC or TCSEC. Both of these look at multiple functions to determine a security level. Anti-Virus applications are only one of several parts of a total system, which would be evaluated under these more formal programs."

The *NCSA* scheme has not been without problems. A certification scheme for the Apple Macintosh platform which was prematurely promoted had no documented test methodology or criteria; we are told it has been discontinued. *NCSA* "Approval" was briefly promoted as a less stringent form of testing, requiring products to pass certain limited tests. This has also been discontinued and the information regarding the "Approval" has been removed from their WWW Site. *NCSA* has provided statements relative to meeting certain limited test criteria for at least one company; the claims have been publicly disputed by industry experts, and we have found the claims to be technically invalid.

However, the PC portion of the scheme developed during 1995 remains viable. Some anti-virus experts have voiced concern over the direction of the scheme, as it is no longer under the direct supervision of an anti-virus specialist. However, Joe Wells, developer of the Wildlist, has agreed to act as an off-site overseer to the testing methodology and maintainer of the virus library. Wells is a recognized industry leader in the field of anti-virus research. The future direction of the scheme remains to be seen; however, according to Martin, "*NCSA* is working with Joe Wells, and the AVPD, to determine any modifications in direction for the current testing scheme. *NCSA* has received requests to perform more formal false alarm testing, to test "TSR" type background protection, and to test repair capabilities of products. Any future changes will be discussed with AVPD before implementation, and would be implemented with a several month lead time." It is the opinion of these authors that anti-virus tests should be performed by specialists with considerable experience in testing. While Wells' qualifications are excellent, the fact remains he is not on-site. This could present problems in test administration and interpretation.

Secure Computing Checkmark, from West Coast Publishing, claims to be a quick, up-to-date, and inexpensive scheme which product developers may use to show independent verification of detection abilities of products. It is hoped that the scheme will provide developers with a way to support detection claims by referring to their independent third-party tests, and provide users with a way to know products meet a minimally acceptable criteria for virus detection. The author of the scheme, Paul Robinson, editor of *Secure Computing*, states that the purpose is to add value back into the industry and to provide benchmarks in the context of evaluating claims. "As reviewers and testers we need to be very transparent. This extends to methodologies; we are telling people exactly how we are testing what we are testing, there is no room for impurity in the test." The scheme is still under development, and appears from the information available to promote the testing of products using documented methodology and criteria. Currently, plans include using the Wildlist as a source for selection of ITW samples; however, identification of included viruses does remain at the discretion of the *Checkmark* administrator. The testing list is to be made available three months prior to the test. Testing is planned quarterly, and will be made of the scanner portion of products only. Vendors will pay an evaluation fee. The fee varies depending on the number of platforms evaluated. The scheme appears to be developing along the same lines as the new *NCSA* scheme in that no vendor will be given exact samples of missed viruses, but rather replicants.

One of the benefits of this scheme is that the methodology is clearly documented and has been distributed to interested parties. However, as the scheme is still in its draft phase, it remains to be seen how widespread acceptance of the standard will be. The documents relating to the scheme furnished to the authors show promise, but only time will tell which direction the final scheme will take.

The Academic Evaluator

Another useful source of information is the Academic Evaluator. Good examples of the type of tests carried out by such evaluators are those by Vesselin Bontchev, formerly of the Virus Test Center (VTC) at the University of Hamburg. The principal advantage with these tests is that the test metrics and methodology are clearly stated. The results are generally presented in a scientific manner and the reader is left with little doubt about how they were obtained [10, 11]. While the tests are another useful and accurate source of information they are limited in scope. Tests seem to be mainly concerned with overall detection rates. Little or no mention is made of detection of those viruses which are known to be ITW, although this information is usually available to those who are prepared to extract it from the raw test data. One potential flaw is that these tests may be carried out by students, who have limited resources and who are performing work in an academic (learning) environment.

The New ITSEC Approach.

The ITSEC was issued within the European Community in the summer of 1991, as an attempt to provide formal internationally-recognized standards for the evaluation of IT products for use within governments. In the UK, the market for evaluated products has been driven by Government procurement policies, especially in the defense industry. The ITSEC concerns relative to anti-virus product evaluation differ from the United States TCSEC. Whereas TCSEC specifies development assurance criteria, ITSEC requires certification and accreditation activities which assess how the product matches the operational environment; i.e., how the product meets the real world threat posed by computer viruses. While there is yet no formal methodology available on paper, the UK ITSEC Anti-Virus Working Group (AVWG) was kind enough to send us information on the status of the project.

Each ITSEC certification requires that products of a particular Functionality Class meet a certain Security Target, which consists of either a Systems Security Policy containing a statement of the security objectives, threats and necessary countermeasures for the system, or a Product Rationale, which contains a list of a product's security features, the intended method of use and the intended environment with its associated threats. The traditional ITSEC approach may be thought of as a "snapshot" of the developer and the product at any one time. Thus, only the version of the product which is evaluated by the Commercial Licensed Evaluation Facility (CLEF) is certified; certification lapses with the very next version of the software released. Anti-virus software evaluation requires a more dynamic approach.

Furthermore, the traditional ITSEC approach includes an examination of the development environment. Current work seems to indicate that in the case of an anti-virus software package it should be possible to extend this examination to include such issues as how well the company is able to maintain its product. It is not sufficient for a company to demonstrate its ability to detect a certain percentage of all known viruses in any one version of its software: it must be able to show that it has appropriate procedures in place to track the threat, and alter the product accordingly to meet it. Involved in building the certification guidelines are vendors such as Sophos (Sweep), S&S International (Dr. Solomon's Anti-virus Toolkit), McAfee (VirusScan), Authentec (Alan Solomon); magazines *Virus Bulletin* and *Secure Computing*; and The BSI (German ITSEC Certification Body). Currently, the evaluation process is in the developmental phase. The main areas with which the process is concerned are Standard Documentation, Threat Assessment, Virus Attack Techniques, AVWG Virus Collection, Comprehensive Virus Collection, "Advice Documentation", and Certificate Maintenance Scheme.

Standard Documentation relates to the development of ITSEC documentation which defines minimum security functionality and related information such as functionality class, security target and suitability analysis. These are largely product independent and will be provided by the AVWG. The documents will then be evaluated by a CLEF and approved by the Certification Body (CB) for use in subsequent anti-virus product evaluations. These documents are in final drafting phase at this time and the CLEFS are now being selected.

In the original version of this paper, we discussed the need for product performance to be measured not only by running detection tests on virus collections, but by testing each product's ability to defend against the different attack mechanisms already observed as well. This obviously requires the maintenance of a library of virus attack techniques, and a collection of samples which utilize each of these techniques. As we explained, this is far better than current evaluations, where without specialized knowledge it possible to "certify" a product which provides no protection against a particular attack technique. Attack techniques should include memory-resident operation and disinfection problems.

The ITSEC attempts to address this area in anti-virus product evaluation by proposing to measure the product's performance against the threat not by running and maintaining a large collection of all viruses, but by testing extensively against those viruses which are known to be ITW, and also against a range of different attack strategies. Thus, the tests should reflect not only the product's ability to defend against those viruses which are ITW, but also against the known threat (by evaluating the product's ability to defend against the different techniques used by viruses) and the future threat (by evaluating the developer's ability to track a rapidly changing threat and update the product to deal with it). Currently, the plan is to feed the assessments into the evaluation process, using reports of incidents, Joe Wells' Wildlist figures, and other available report information. This solution can lead to possible problems as new threat types may be as yet unanalyzed, and the virus itself is not ITW. There is no guarantee as to the time sequence that a virus may be found to exist, be found in the wild, obtained and analyzed by an evaluation or certification service, and its threat type documented. This is illustrated by the recent spate of macro viruses, where there was a noticeable lag between the discovery of the virus (that is, the creation of the threat type), and the implementation of detection and prevention on the part of some developers.

A Virus Attack Techniques Encyclopedia (VATE) has been developed under contract by the AVWG. This is intended to detail all known techniques used by viruses. It is a dynamic document. The VATE will be used to direct morc detailed analysis and testing of products; it is a limited distribution document.

Product manufacturers must of course include detection for all viruses, whether or not they are found ITW, bccausc thc mere existence of a virus constitutes a threat to users. For this reason, it may be prudent to have both entire libraries and attack stratcgy suites. The AVWG currently is in the process of establishing a virus collection to support the evaluation process. There is no intention to make this comprehensive, as they have neither the staff nor the expertise to maintain a comprehensive collection. Rather, the collection will contain ITW viruses and examples of viruses illustrating the range of attack techniques covered by the VATE. The anticipated number of viruses is 100-1000. Advice on generation of test suites is still being received. The source of comprehensive virus collection to be used during evaluations is under discussion within the AVWG at the time of writing.

In addition to formal ITSEC documentation, the AVWG recognizes the need for a considerable volume of supporting documentation. There will be the current characterization of the threat (In the Wild list, VATE and virus test suites); general advice to evaluators on how to do product testing; information on special cases; the interpretation of test results; criteria for acceptance. Some of this may be incorporated into the existing UK Manual of Evaluation (UKSP05). Advice documentation to vendors may be included into the UK ITSEC Developers guide (UKSP04). The advice documentation is presently being written, but cannot be completed until the formal requirements such as Functionality Class and Security Target are finalized.

In summary, the functionality tests related to virus detection would be comprised of tests of four types:

1. Common Viruscs (determined from AVWG threat tracking)
2. ITW Viruses (determined from AVWG threat tracking, Joe Wells' In the Wild list, other information from the AV community)
3. Virus Attack Techniques (from the VATE)
4. Tests against a "comprehensive" virus collection approved by the AVWG.

An increasing level of rigor would be applied and associated with the commonality of the virus or observed technique, i.e. weighted testing. The current plan is to perform tests with 1&2 listed concurrently and cumulatively and to require a 100% score to pass. Thc current strategy for zoo testing is 90% for a passing score, based on industry input.

The evaluating body would operate in close contact with the developer of the product currently under evaluation. This means that developers will have to demonstrate that not only are they up to date with the current threat, but that sufficient procedures are in place to monitor the threat as a function of time and update the software to match it. This "vendor evaluation" is something which almost all other evaluations of anti-virus software do not include, and is one of the biggest benefits of the proposed AVWG ITSEC approach. It is also one of the areas which appears to meet with the most resistance within the USA. Another concern which has been cited [12] is regarding the sharing of information between CLEFs: "Even though the UK require that all techniques and lessons learnt from evaluations be documented at the end of an evaluation and made available to the UK evaluation community, it is felt that CLEFs prepare this information from a position of non-disclosure of information which is of a proprietary interest to them. There is concern in the US that UK evaluation, by virtue of their commercial nature, do not encourage the sharing of evaluation techniques amongst the evaluation community".

Finally, there are problems with issues of legal liability. Whereas German law demands someone be liable for failure in certified products, the United States makes specific disclaimers assuming no responsibility. Drawing from Borrett [12], we find "the political implications of legal liability for Europe and North America merits further investigation. In the interim, it may suffice to place an appropriate caveat alongside any US evaluated products which appear in UK Certified Product List publications."

Additionally, it is very difficult to estimate the cost of an evaluation without actually submitting a product: the amount of work needed to be done could vary with the claims made by the developer and the precise nature of the anti-virus software. Unfortunately, it is still too early for a precise estimate of the costs: until a functionality class has been formally defined. The ITSEC/AVWG hopes to have the evaluation process functional by the end of 1996.

Summary of the Problems

Thus, we have shown that none of the groups above can perform anti-virus software evaluations which fit all the needs of those who are attempting to make a purchasing decision.

Aside from the problems which are unique to each tester, we have discussed several difficulties which are shared between almost all anti-virus software reviewers, testers, evaluators, and certifiers:

- Choice of virus test-set. Does the evaluator have the technical skills necessary to maintain and sort a large virus test-set? Using a scanner to determine infected/non-infected state of files is clearly unacceptable. Viruses must be replicated, and first generation samples are unacceptable. The problems of maintaining a clean, well-ordered virus test-set are discussed further by Bontchev [13]. Creation of the test suite includes the minimum of the following (some taken from [14]):
    - Replication of live boot viruses on all media (5.25 360k diskettes, 5.25 1.2 MB diskettes, 3.5 720k diskettes, 3.5 1.44 MB diskettes, HD master boot record and HD DOS Boot sector).
    - Replication of live file viruses including COM files consisting of normal files, files beginning with JMP instruction, COMMAND.COM, file with many NOPs, files infected multiple times; EXE files consisting of normal files, files with 0 and multiple relocations, Windows applications, compressed files etc.
    - Replication of polymorphic viruses of low polymorphism consisting of 10-10,000 replicants and high polymorphism consisting of at least 10,000 samples (100,000 is not unheard of).
    - Replication of companion viruses, macro viruses and multi-partite samples onto appropriate hosts.
- Time involved. Generation of the test suites described above is dynamic, as new viruses are found daily. Additionally, testing is another time consuming process. Testing includes but is not limited to cleaning of memory and media, checking of system integrity, infection of the victim files and/or boot sectors, checking replication potential of the replicants, scanning and report generating.
- Bias. Is the evaluator in any way associated with one of the products which is reviewed? Were the samples obtained from a particular vendor?
- Which aspects of the product have been tested? Were the test results weighted, and if so, how?
- Which tests measure the efficacy of the disinfection routines, the efficiency of memory scanning or the problem of false positives, user interface and documentation; how were they conducted and how were the results interpreted?
- Has the product been tested for compatibility with your system/network and are additional tools provided?
- Has company support/tech support been evaluated? Areas of company support which should be evaluated are response time via telephone and electronic media, completeness of information provided and follow-up.

In summary, the problems with anti-virus product evaluation are many. The ITSEC approach provides some suggestions as for how we can adapt and use their fundamental approach to evaluating products, but, as we have seen above, even this is not a complete system.

Conclusion

We have examined the current evaluation methods applied to anti-virus software, and demonstrated that at best they only cover some of the areas which a complete evaluation of a product should cover. We believe that the current plans for anti-virus software evaluation in the ITSEC will address many of these issues, and that when the system is fully operational it will provide the prospective purchaser with some guarantee of software functionality, and moreover some measure of the developer's commitment to continue to meet a rapidly changing threat. We note that the ITSEC methods are not a cure all, and that even if plans of the AVWG are implemented, there are still areas which do not appear to be satisfactorily addressed.

While we recognize the problems of the ITSEC, we believe that the underlying methodology is sound, and that by drawing from the positive addition of new forms of functionality testing and product assessment, we are hopeful that in the near future anti-virus product evaluators of all types will have a more solid knowledge base from which to draw.

We believe that not only is it impractical to perform all aspects of product evaluation in-house, but that doing so can be directly damaging, as it is possible to select a product for entirely the wrong reasons. Thus, the reader is urged to use a wide variety of sources of information. Much of the information outlined above can be obtained at little or no cost; by understanding the strengths and weaknesses of each different evaluation you are in a position to extract figures which are relevant to determining which product is most suitable for your company.

It is still necessary to cull information from a number of sources to select a product which not only fulfills the functionality which is required by your policy (speed, transparency, cost), but also provides an adequate defense against the threat (virus detection). This can only be done by carefully considering your anti-virus policy and creating a list of requirements which your chosen product must fulfill. The first criterion remains "how well does the product detect viruses you are likely to encounter".

Keep in mind, that as the user of any anti-virus product evaluation service, you should be encouraged to contact the evaluator to get any relevant information not contained within the review [7]: only by recognizing the strengths and weaknesses of existing product evaluation schemes can we hope to use the currently-available information to our advantage when attempting to choose the "right" product for your environment.

Bibliography

[1] In Laine [3].

[2] Richard Ford, Sarah Gordon. *Real-World Anti-Virus Product Review and Evaluation,* IVPC 95 Conference Proceedings.

[3] Kari Laine. *The Cult of Anti-Virus Testing.* EICAR 1994 Conference proceedings.

[4] Sarah Tanner. *A Reader's Guide to Reviews*, Virus News International, November 1993.

[5] Sarah Gordon. "*Is a Good Virus Simulator Still a Bad Idea*?" Preprint.

[6] Virus Bulletin. *Disinfection: Worth the Risk?,* September 1994.

[7] Sarah Gordon. *Evaluating the Evaluators*, Virus News International, July and August 1993.

[8] NCSC, *Introduction to Certification and Accreditation,* Rainbow series, NCSC-TG029, January 1994.

[9] Richard Ford. Private communication. 1995.

[10] Marko Helenius. *Anti-Virus Scanner Analysis by Using The 'In the Wild' Test Set.* EICAR 1994 Conference proceedings.

[11] VTC. Anti-virus scanners test protocol. Virus Test Center, University of Hamburg, Germany.

[12] Alan Borrett. *A Perspective of Evaluation in the UK Versus the US.* Proceedings 18th National Information Systems Security Conference. Baltimore, Maryland. October 1995.

[13] Vesselin Bontchev. *Analysis and Maintenance of a Clean Virus Library*, Virus Bulletin Conference Proceedings , Amsterdam, 1993.

[14] Vesselin Bontchev, Klaus Brunnstein, Wolf-Dieter Jahn. *Towards Antivirus Quality Evaluation.* Virus Test Center, Faculty for Informatics. University of Hamburg, Germany. From the Proceedings of the 3rd EICAR Conference, Munich Germany. December 1992.

About the Authors:

Sarah Gordon is Security Analyst for Command Software Systems, where she works in Research and Development, maintaining the virus library. Dr. Richard Ford is Technical Director for Command Software Systems, and works in the area of product testing. Both Ms. Gordon and Dr. Ford have extensive experience in testing anti-virus products and have published numerous articles on computer viruses and other computer security topics. They may be reached respectively at sgordon@dockmaster.ncsc.mil and rford@commandcom.com.

Acknowledgements: Megan Alexander, Command Software Systems.

Anti-Virus Product Evaluation in the Real World

The current state of affairs

Sarah Gordon

Richard Ford

Command Software Systems

Who's Who?

- Friends
- Employees
- Tech Support Staff
- Independent Reviewers
- Magazines
  - General
  - Virus/Security
- Commercial Evaluators
- Academic Testers
- Executive Summarizers
- Governmental Bodies
- Vendors
- ITSEC AVWG

FRIENDS AND OTHERS

- Friendly Advice
  - "It works great"
  - "I've never had a virus"
  - "It's fast!
- Employees
  - "I love to help out at work!"

TECHNICAL SUPPORT STAFF

- "I'm technical"
  - Novell, UNIX, VMS
- "I know about viruses"
  - Usenet, World Wide Web
- "I have equipment here!"
  - uhhhh...*which* equipment?

Magazines

GENERAL

- Virus collections
  - vendor, bbs, ftp, www, CD-ROM, simulator
- Testing competency
  - flawed tests
- Legal liability
- Bias

VIRUS/SECURITY

- Virus collections
  - usually good
- Testing competency
  - competent
  - documented
  - usually well interpreted
- Bias

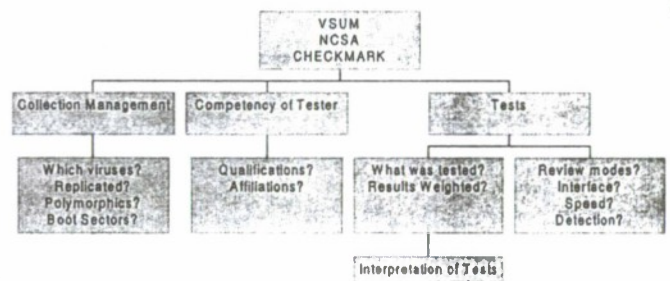
INDEPENDENT EVALUATORS

- Who
  - qualifications
  - affiliations
- Where
  - Virus-L
  - FidoNet

Scholars and other Strangers

- Academics
- Executive Summarizers
- Vendors

COMMERCIAL EVALUATORS



ITSEC AVWG

- Common viruses
- ITW Viruses
- VATE
- Tests Against Industry Standard Collection

using CLEFs

Problems common to all

- Choice of test suite
- Time involved
- Bias
- Limited Functionality Testing
  - compatibility
  - scanner, tsr, disinfection
- Evaulation of tech support

Suggestions

- Realize there is not yet one complete solution
- Decide who will evaluate software
  - be aware of all influences
- Designate what will be evaluated
- Ascertain how it will be evaluated
  - gather information from specialists
  - virus/Security Specialist Publications
  - NCSA/Checkmark

Caveats

- Do not increase your organization's vulnerabilities!
  - no in-house "tests" using simulators, CD-ROMS, FTP site, or WWW viruses!
  - weigh advice from "experts" carefully
- Do not expect more from your staff than they can reasonably be expected to provide!

SPOCK

SECURITY PROOF OF CONCEPT KEYSTONE

James McGehee
COACT, Inc.
9140 Guilford Road, Suite L
Columbia, Maryland 20146

In 1992 representatives from the vendor community and National Security Agency believed that emerging security products could provide some security solutions within a given architecture. The goal of the group was to seek out security products and demonstrate their usefulness within government system architectures. This goal was the keystone for the established program called Security Proof-of-Concept Keystone (SPOCK).

The SPOCK program is a joint government and industry forum sponsored by the National Security Agency to demonstrate security features of commercial and government products that can support dependable security architectures. The activity provides a forum for government users and security technology providers to share information on security requirements, emerging technologies, and new product developments. Integrators and product developers are afforded opportunities to share new solutions, identify government developed technology available for commercial use, and prototype commercial-off-the-shelf products in government sponsored test beds. The SPOCK forum meets monthly to share information about emerging architectures, secure products, security requirements, threats, standards and building codes. SPOCK members include representatives from the National Security Agency, military services, government services, including agencies outside of the Department of Defense, and industry to include integrators, and product developers. Product developers, contractors and test bed clients participating in SPOCK initiatives are permitted and encouraged to volunteer time, materials, and personnel according to the perceived value of the initiatives. To be a member and to participate in the group, representatives from government and industry organizations should have security awareness, be involved in communications products or services (including software), understand that security integration does affect change in products and services, be an individual or organization who targets Information Security as a necessary technology, and be willing to share information and resources to improve our knowledge base and ability to implement security products.

The purpose of SPOCK is to:

a) Demonstrate that current certified products can provide a measure of systems security.

b) Determine if any uncertified system components can be used to improve a secure system.

c) Define products that can support secure architectures.

d) Define the risks in using these secure architectures.

e) Showcase technology--not develop it

The group has developed a capability to do testing and proof-of-concept demonstrations on products within given architectures both in the laboratory and in operational network settings. The proof-of-concepts are designed to independently verify the accuracy of vendor claims about the security of their products.

The SPOCK program makes use of existing laboratories and contract vehicles. It provides a forum for government and industry to have a continuing dialogue toward solving network security requirements. In addition to testing and proof-of-concept demonstration opportunities, it also provides an archive of completed proof-of-concept reports on system architectures and products with security features and policy for members and network architects to use. At the monthly meetings briefings are given by government representatives that describe architectures, requirements, or new government developed security technology. From commerce representatives, briefings are presented on new security products, implemented security architectures, or commercial sector requirements.

SPOCK participation is voluntary. The focus is Information Security. Presentations and proof-of-concepts are proposed and presented by any participant.

Presentations and proof-of-concepts are proposed by the forum membership. A proof-of-concept demonstration begins with identification of Vendor Claims and a sponsored architecture to be tested. When a proposed proof-of-concept is accepted by the SPOCK Chairman (a National Security Agency member), a team is formed. This team is composed of volunteer forum members who are interested in the proof-of-concept or who can contribute resources (i.e technical support, hardware, software, test equipment, connectivity, etc.). A test plan is written and agreed to by all participants in the proof-of-concept demonstration. The test plan focuses on the vendors claim package. In addition, performance tests are applied when possible. The SPOCK integration contractor coordinates support between team players, supervises the demonstration and test activities and publishes the final test report. A draft report is written, reviewed by the test team and approved by the SPOCK chairman. The report is then published and distributed to interested participants. All of the reports are controlled. They are not classified. SPOCK reports can be requested through the integration contract:

COACT, Inc.
9140 Guilford Road, Suite L
Columbia, Maryland 21046
Phone: 301-498-0150
Fax: 301-498-0855

The following are some examples of proof-of-concept test plans and reports:

1). BLACKER Front End LAN, Document No.1600383, 14 December 1993

2). Raptor, Eagle/Eaglet Test Plan, Document No. 1600390, October 1993

3). Raptor Eagle/Eaglet, Test Report, Document No. 1600393, February 1994

4). Filter Router Test Plan - Phase I, Document No. 1600386, November 1993

5). Filter Router Test Report, Executive Summary, Document 1600411, April 1994 (3COM, Alantec, CISCO, Network Systems, Proteon, and Wellfleet)

6). Buttress Test Report, Document No. 1600424, 13 June 1994 (a successful joint Air Force, Navy, Sprint, SPOCK initiative to provide off-board imagery and emitter information to an aircraft in a timely fashion to support targeting of non-line-of-sight targets for tactical air strikes)

7). Network Security Router, Performance and Security Test, Document No. 10504, 29 March 1996

The latter was the most recent proof-of-concept conducted by the SPOCK Program to validate vendor claims of performance and security goodness of the Network Systems Corporation's, Security Router. Participants in the proof-of-concept were the Air Force Space Command Space Warfare Center, the Army Battle Command Battle Laboratory, the Internal Revenue Service, NSA/V2, NSA/Y4, Network Systems Corporation, and COACT,Inc. The Internet was used as a connecting medium between the test nodes. Performance testing and mandatory access control (MAC) testing was performed at and by the IIT Research Institute (an Internal Revenue Service federally funded research and development contractor). Penetration testing was conducted by

NSA/C44 personnel. The tests were monitored by SPOCK participants. The result of tests performed showed that when configured properly, the router would provide highly reliable and secure communications across an unsecured network, and that data could be passed at speeds in excess of 1 Mbps. Applied attempts to penetrate the network from outside of trusted enclaves were unsuccessful. The following is an example of Vendor Claims.

EXAMPLES OF VENDOR SECURITY CLAIMS

Network Attack Protection

Selectively permit traffic through the router

Protect against IP level spoofing

Provide audit of attack violations

Prevent and audit unauthorized protocols

Prevent and audit unauthorized network service applications

Prevent and audit fragments from entering networks

Prevent and audit source routed packets

Data Privacy

Encrypts data transmitted by the router at 1 Mbps

Prevents access to public key information during exchange

Detect and audits replay attacks

Authenticates communicating routers

Mandatory Access Control

Selectively allows traffic based on RIPSO labels

Assign default labels to unlabeled datagrams

Routes datagrams based on RIPSO labels

Encrypts datagrams based on RIPSO labels

As a result of completing a proof-of concept under the auspices of SPOCK, a Memorandum is issued and signed by the Chief of NSA/V2 as the SPOCK Chairman.

Memorandum

To:	SPOCK Consortium
CC:	
From:	Bill Marshall
Date:	April 30, 1996
Subject:	SPOCK Demonstration Report - NSC Security Router

The SPOCK Consortium, as part of its continuing goal to explore INFOSEC commercial solutions and enabling technologies, is pleased to issue this demonstration report on the NSC Security Router.

The report validates vendor claims about security functionality of its product in 'warfighter' architectures. Validation tests were conducted over a two month period. The report provides automated information system integrators and architects an overview of the product security functionality in government architectures.

Bill Marshall
Chief V2 NSA SPOCK Chairman

At each monthly SPOCK meeting, discussions, briefings and sharing information takes place. The following are examples of previous presentations:

a) Common Criteria (V2)
b) Sterling Software Secure Network (Sterling)
c) DirecPC (Hughes Information Technology Systems)
d) Shipboard Network Integration (Lockheed/Martin)
e) Dockmaster II
f) C4 Attack Center (C44)
g) MISSI Certificate Architecture (NSA/X33)
h) NSC Secure Router (NSC)
i) ATM Networking (NSC)
j) Virtual Campus (NSA/Y44)
k) Pathkey (Paralon)
l) Joint Interoperability Test Center Capability (JITC)
m) Joint Warfighter Interoperability Demonstrations (NSA/V2)
n) INFOGUARD, ATM Cell Encryptor (Cylink and GTE)

For efficient response to proof-of-concept proposals, SPOCK takes advantage of existing laboratories and networks. These resources can be in government or commercial sites. Current sites are the Space Warfare Center at Falcon Air Force Base, Colorado, the Army Battle Command Battle Laboratory at Fort Gordon, Georgia, IITRI in Lanham, Maryland., and the National Security Agency at Fort Meade, Maryland.

An important segment of the SPOCK program is its commitment to support the Warfighter effort. SPOCK has been introduced to the Joint Warfighter Interoperability Demonstration (JWID) program managers. Discussions are continuing on ways for SPOCK to support the JWID demonstrations.

It should be noted that the SPOCK program is not intended to, nor does it compete with programs such as the Trusted Computer Security Evaluation Criteria (Orange Book), the Common Criteria Program, the National Institute of Standards and Technology initiatives and programs, or the Multilevel Information Systems Security Initiative. SPOCK supports these formal type of initiatives by providing data that gives customers, developers and evaluators an early view of the product/system security attributes. This data can support decisions by the customer as to whether the system fulfills or has potential to fulfil their security needs. This data helps the developer determine the state of his security functions and assurances. It can help the developer determine whether the product is ready to proceed with a formal evalaution or does it need more tweaking. Finally, this data can support the evaluator when forming judgements about the conformance of the product/system to targeted security requirements.

SPOCK provides a low cost, and quick look at security products within a specific architecture. The SPOCK proof-of-concept reports provide empirical information to network architects and accreditors. This data can help them to make informed decisions concerning their architectures and products that can be effectively used in their architectures. Some valued added features of the SPOCK Program include:

a) Evaluated, certified, or endorsed products can be prototyped in test bed configurations that may be different from those for which the product was originally reviewed.

b) Products can be prototyped to determine the usefulness of uncertified or untrusted products and solutions in client architectures.

c) Information Security products, processes, policies and technologies can be reviewed in test architectures.

d) Test beds can be used to prototype innovative Information Systems Security Engineering (ISSE) techniques.

e) Independent validation of Product developer claims

f) Supports accreditation and certification initiatives

SPOCK continues to focus on emerging security technologies. Vendor claims have been received for the IRE Fortezza Modem (Industrial Research Engineering), and the INFOGUARD ATM Cell Encryptor (Cylink and GTE). Development of test architectures and test plans are on-going. Other potential proof-of-concepts include the Network Systems ATM Encrypting Router, and the DirecPC (Hughes Information Technology Systems) which provides a global broadcast capability to include encryption.

In summary, SPOCK has been successful. The monthly meetings and the proof-of-concept demonstrations have provided useful information to the vendor for design, development and product improvements. The developers of security products have the opportunity meet potential customers. Integrators have the opportunity to learn about new products for security solutions. The SPOCK customers such as accrediting authorities have been provided valuable data needed to assist in making decisions about security products usefulness.

References:

1. SPOCK CONCEPT OF OPERATIONS, Document No. 5400001, Revision 7, August 1995

SPOCK

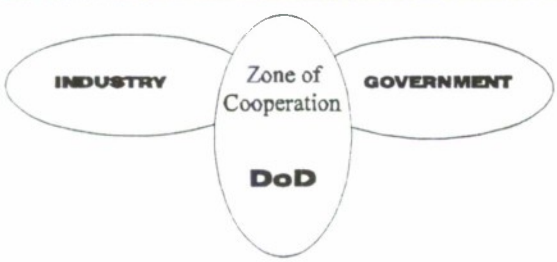
Security Proof of Concept Keystone

James McGehee
COACT, Inc.
301-498-0150

WHAT IS SPOCK ?

- SPOCK is a consortium of industry developers and
- Governnment integrators interested in exploring INFOSEC
- Commercial solutions and enabling technologies

SPOCK MEMBERSHIP



WHAT DOES SPOCK DO ?

- SPOCK holds monthly meetings to discuss security products and systems that help to secure architectures
- SPOCK demonstrates security in "Warfighter" and government architectures

SHARING OF INFOSEC TECHNOLOGY

- Common Criteria
- DOCKMASTER II
- PATHKEY
- C4 Attack Center
- NSC Secure Router
- Virtual Campus
- MISSI
- JITC
- Sterling
- JWID
- INFOGUARD
- Shipboard Networks
- DirecPc
- Fortezza Modem
- Intelligent Agents
- Crypto SmartDisk

SPOCK DEMONSTRATIONS

- Validation of Vendor Security Claims
- Developer Submits Security Claims
- SPOCK Validates the Claims by Testing, Writing and Distributing the Reports

VENDOR SECURITY CLAIMS

- NETWORK ATTACK PROTECTION
- MANDATORY ACCESS CONTROL
- DATA PRIVACY

SPOCK DEMONSTRATION PARTICIPANTS

- Space Warfare Center
- NSA - Y4, C4, V2, G04
- Internal Revenue Service
- Battle Command Battle Lab

WHY USE SPOCK ?

Consumers: Help to decide whether a product or system can be used prior to an Evaluation or Certification

Developers: Supports preparation for a formal evaluation/certification process

Evaluators: Provides data to assist in forming judgements about conformance of product

SPOCK BENEFITS

- Independent Validation of Developer Security Claims
- Rapid Security Technology Review
- Teamed Demonstration Efforts
- Supports Accreditation and/or Certification Initiatives

POINTS OF CONTACT

SPOCK Program Manager - Terry Losonsky, V21 (NSA) 410-859-6091

SPOCK "Navigator" - CPT. Jay Artiaga, V21 (NSA) 410-859-6091

SPOCK Support Contract - COACT,Inc. 301-498-0150

Use of A Taxonomy of Security Faults

Taimur Aslam, Ivan Krsul, and Eugene H. Spafford
COAST Laboratory
Department of Computer Sciences
Purdue University
West Lafayette, IN 47907–1398
{aslam,krsul,spaf}@cs.purdue.edu

July 9, 1996

Abstract

Security in computer systems is important so as to ensure reliable operation and to protect the integrity of stored information. Faults in the implementation of critical components can be exploited to breach security and penetrate a system. These faults must be identified, detected, and corrected to ensure reliability and safeguard against denial of service, unauthorized modification of data, or disclosure of information.

We define a classification of security faults in the Unix operating system. We state the criteria used to categorize the faults and present examples of the different fault types.

We present the design and implementation details of a prototype database to store vulnerability information collected from different sources. The data is organized according to our fault categories. The information in the database can be applied in static audit analysis of systems, intrusion detection, and fault detection. We also identify and describe software testing methods that should be effective in detecting different faults in our classification scheme.

1 Introduction

Security of computer systems is important so as to maintain reliable operation and to protect the integrity and privacy of stored information.

In recent years we have seen the development of sophisticated vulnerability databases and vulnerability exploitation tools by the so-called "computer underground". Some of these tools are capable of automating the exploitation of vulnerabilities that were thought to require considerable expertise, including IP and DNS spoofing. These tools are freely and widely available, and pose a significant threat that cannot be ignored. The celebrated Kevin Mitnick is an example of a vandal who used such tools and databases to penetrate hundreds of computers before being caught [17]. Although Mitnick was an expert at exploiting VMS security holes, it is widely believed that his knowledge of Unix was limited and that he was provided, by a source unknown, with ready-made tools of considerable complexity [30].

With the widespread use of computers, and increased computer knowledge in the hands of people whose objective is to obtain access to unauthorized systems and resources, it is no longer possible or desirable to implement security through obscurity [16].

To ensure that computer systems are secure against malicious attacks we need to analyze and understand the characteristics of faults that can subvert security mechanisms. A classification scheme can aid in the understanding of faults that cause security breaches by categorizing faults and grouping faults that share common characteristics.

2 Related Work

Existing fault classification schemes are not suitable for data organization because they do not clearly specify the selection criteria used. This can lead to ambiguities and result in a fault being classified in more than one category.

The Protection Analysis (PA) Project conducted research on protection errors in operating systems during the mid-1970s. The group published a series of papers, each of which described a specific type of protection error and presented techniques for finding those errors. The proposed detection techniques were based on pattern-directed evaluation methods, and used formalized patterns to search for corresponding errors [13]. The results of the study were intended for use by personnel working in the evaluation or enhancement of the security of operating systems [10].

The objective of this study was to enable anyone with little or no knowledge about computer security to discover security errors in the system by using the pattern-directed approach. However, these method could not be automated easily and their database of faults was never published. The final report of the PA project proposed four representative categories of faults. These were designed to group faults based on their syntactic structure and are too broad to be used for effective data organization.

The RISOS project was a study of computer security and privacy conducted in the mid-1970s [6]. The project was aimed at understanding security problems in existing operating systems and to suggest ways to enhance their security. The systems whose security features were studied included IBM's OS/MVT, UNIVAC's 1100 Series operating system, and Bolt Beranek and Newman's TENEX system for the PDP-10. The main contribution of the study was a classification of integrity flaws found in the operating systems studied.

The fault categories proposed in the RISOS project are general enough to classify faults from several operating systems, but the generality of the fault categories prevents fine-grain classification and can lead to ambiguities, classifying the same fault in more than one category.

Carl Landwehr et al. [24] published a collection of security flaws in different operating systems and classified each flaw according to its genesis, or the time it was introduced into the system, or the section of code where each flaw was introduced. The taxonomy proposed, unfortunately, is difficult to use for unambiguous classification because the categories are too generic and because it does not specify a clear classification criteria.

Brian Marick [25] published a survey of software fault studies from the software engineering literature. Most of the studies reported faults that were discovered in production quality software. Although the results of the study are insightful, the classification scheme provided is not suitable for data organization and unambiguous classification.

Although classical software testing techniques are not strictly concerned with a taxonomy of software flaws, we must pay close attention to them because fault classification schemes must classify faults detected using these methods.

Boundary Condition Errors:
Boundary Value Analysis (BVA)can be used to design test cases for functional testing of modules. BVA ensures that the test cases exercise the boundary conditions that can expose boundary condition errors [26]. In addition to functional testing, mutation testing can also be used to detect boundary conditions by designing appropriate language dependent mutants [7, 12, 31, 14].

Domain analysis can be applied to detect boundary condition errors. Domain analysis has been studied with two variables and examined with three variables [19, 5]. The main disadvantage of domain testing is that it can only be applied to a small number of variables as the difficulty of selecting test cases becomes increasingly complex. In an experiment by Howden, path analysis revealed the existence of one out of three path selection errors [18].

Input validation Errors: These errors result when a functional module fails to properly validate the input it accepts from another module or another process. Failure to validate the input may cause the module accepting input to fail or it may indirectly cause another interacting module to fail.

Syntax testing can be used to verify that functional modules that accept input from other processes or modules do not fail when presented with ill-formatted input.

Path analysis and testing can be applied to detect scenarios where a certain execution path may be chosen based on the input. In an experiment conducted by Howden, path testing revealed the existence of nine out of twelve computation errors.

Access Validation Errors: Path analysis can be used to detect errors that result from incorrectly specified condition constructs. Branch and Relational Operator testing (BRO) is a test case design techniques that can aid in the design of test cases that can expose access validation errors.

Failure to Handle Exceptional Condition Errors: A security breach can be caused if a system fails to handle an exceptional condition. This can include unanticipated return codes, and failure events.

Static analysis techniques such as inspection of design documents, code walk-throughs, and formal verification of critical sections can be used to ensure that a system can gracefully handle any unanticipated event. Path analysis testing can also be performed on small critical sections of code to ensure that all possible execution paths are examined. This can reveal problems that may not have been anticipated by the designers or overlooked because of complexity.

Environment Errors: These errors are dependent on the operational environment, which makes them difficult to detect [31]. It is possible that these vulnerabilities manifest themselves only when the software is run on a particular machine, under a particular operating system, or a particular configuration.

Spafford [31] used mutation testing to uncover problems with integer overflow and underflow. Mutation testing can be used to design test cases that exercise a specific set of inputs unique to the run-time environment. Path analysis and testing can also be applied to sections of the code to ensure that all possible inputs are examined.

Synchronization Errors: These are introduced because of the existence of a timing window between two operations or faults that result from improper or inadequate serialization of operations. One possible sequence of actions that may lead to a synchronization fault can be characterized as [22]:

1. A process acquires access to an object to perform some operation.
2. The process's notion of the object changes indirectly.
3. The process performs the operation on the object.

Mutation testing can be used to detect synchronization faults in a program. To detect faults that are introduced by a timing window between two operations, a `trap_on_execution` mutant can be placed between these two operations. The mutant terminates execution of the program if certain specified conditions are not satisfied. For instance, a timing window between the access permission checks and the actual logging in `xterm` could be exploited to compromise security [3]. A mutant for this vulnerability could be designed that terminated execution thus killing the mutant, if the access checks had been completed. This mutant could be placed between the access checks and the logging to detect the race condition.

Mutants can also be designed to detect improper serialization operations. Consider a set of n statement that must be executed sequentially to ensure correct operation. We assume that the statements do not contain any instructions that break the sequential lock-step execution. We can design $(n! - 1)$ mutants that rearrange the order of the n execution statements. These mutants are killed when the mutated program produces a different result than the original program.

Configuration Errors: These may result when software is adapted to new environments or from a failure to adhere to the security policy. Configuration errors consist of faults introduced after software has been developed and are faults introduced during the maintenance phase of the software life-cycle.

A static audit analysis of a system can reveal a majority of configuration errors. Among the various software testing techniques discussed, static analysis is the most effective in detecting configuration errors. The static audit of a system can be automated by using static audit tools such as COPS [15] and Tiger [29] that search a system for known avenues of penetration.

3 Fault Classification Scheme

From the work presented in the previous section, and from our experience working with security faults, we developed a taxonomy of security faults that is more appropriate for data organization. We broadly classify faults as either coding faults or emergent faults.

Although personnel, communication, physical, and operations security also play an essential role in the reliable operation of computer systems, we focus on faults that are embodied in the software.

Coding faults are comprised of faults that were introduced during software development. These faults could have been introduced because of errors in programming logic, missing or incorrect requirements, or design errors [28, 32, 27, 9, 20].

Emergent faults result from improper installation of software, unexpected integration incompatibilities, and when when a programmer fails to completely understand the limitations of the run-time modules. Emergent faults are essentially those where the software performs exactly according to specification, but still causes a fault. Most policy errors can be classified as emergent faults, as can be modular sofware where each module works perfectly but the integrated product does not.

For classification purposes, we abstract each implementation error to a level that will maintain the specific characteristics yet hide the implementation details. This approach is beneficial when classifying faults from more than one programming language.

Our taxonomy of faults is comprised of the following categories:

Coding Faults

- Synchronization errors.
- Condition validation errors.

Emergent Faults

- Configuration errors.
- Environment faults.

3.1 Synchronization Errors

In our taxonomy a fault classifies as a synchronization error if:

- A fault can be exploited because of a timing window between two operations.
- A fault results from improper serialization of operations.

For example, a vulnerability was found in many versions of the **xterm** program which, if exploited, allowed users to create and delete arbitrary files in the system. If **xterm** operated as a setuid or setgid process, then a race condition between the access check permissions to the logging file and the logging itself allowed users to replace any arbitrary file with the logging file [3]. The following code illustrates how the vulnerability would be exploited.

```
# create a FIFO file and name it foo
mknod foo p
# start logging to foo
xterm -lf foo
# rename file foo to junk
mv foo junk
# create a symbolic link to password file
ln -s /etc/passwd foo
# open other end of FIFO
cat junk
```

This error occurs because of a timing window that exists between the time access permissions of the logging file are checked and the time actual logging is started. This timing window could be exploited by creating a symbolic link from the logging file to a target file in the system. As **xterm** runs setuid root, this could be used to create new files or destroy existing files in the system.

3.2 Condition Validation Errors

Conditions are usually specified as a conditional construct in the implementation language. An expression corresponding to the condition is evaluated and an execution path is chosen based on the outcome of the condition. In this discussion, we assume that an operation is allowed to proceed only if the condition evaluated to true. A condition validation error occurs if:

- A condition is missing. This allows an operation to proceed regardless of the outcome of the condition expression.
- A condition is incorrectly specified. Execution of the program would proceed along an alternate path, allowing an operation to proceed regardless of the outcome of the condition expression, completely invalidating the check.
- A predicate in the condition expression is missing. This would evaluate the condition incor-

rectly and allow the alternate execution path to be chosen.

Condition errors are coding faults that occur because a programmer misunderstood the requirements or made a logic error when the condition was specified.

In our taxonomy, a fault classifies as a condition error if one of the following conditions is missing or not specified correctly:

Check for limits. Before an operation can proceed, the system must ensure that it can allocate the required resources without causing starvation or deadlocks. For input/output operations, the system must also ensure that a user/process does not read or write beyond its address boundaries.

Check for access rights. The system must ensure that a user/process can only access an object in its access domain. The mechanics of this check would differ among different systems depending on how access control mechanisms are implemented.

Check for valid input. Any routines that accept input directly from a user or from another routine must check for the validity of input. This includes checks for:

- Field-value correlation.
- Syntax.
- Type and number of parameters or input fields.
- Missing input fields or delimiters.
- Extraneous input fields or parameters.

Failure to properly validate input may indirectly cause other functional modules to fail and cause the system to behave in an unexpected manner.

Check for the origin of a subject. In this context, subject refers to a user/process, host, and shared data objects. The system must authenticate the subject's identity to prevent against identity compromise attacks.

In Unix, `/etc/exports` specifies a lists of trusted remote hosts that are allowed to mount the file system. In SunOS 4.1.x, if a host entry in the file was longer than 256 characters, or if the number of hosts exceeded the cache capacity, a buffer overflow allowed any non-trusted host to mount the file system [4]. This allowed unauthorized users read and write access to all files on a system. This error occurred because the system failed to check that it had read more than 256 characters or that it had exhausted the cache capacity.

Another example is the **uux** utility in Unix. This utility allows users to remotely execute a limited set of commands. A flaw in the parsing of the command line allowed remote users to execute arbitrary commands on the system [11]. The command line to be executed was received by the remote system, and parsed to see if the commands in the line were among the set of commands that could be executed. **uux** read the first word of the line, and skipped characters until a delimiter character (;,^, |) was read. **uux** would continue this way until the end of the line was read. However, two delimiters (&, ') were missing from the set, so a command following these characters would never be checked before being executed. For example, a user could execute any command by executing the following sequence.

```
uux remote_machine ! rmail anything & command
```

In **uux** the command after the "&" character would not be checked before being executed. This allowed users to execute unauthorized commands on a remote system. This error occurred because **uux** failed to check for the missing delimiters.

3.3 Configuration Errors

The configuration of a system consists of the software and hardware resources. In our taxonomy, a fault can be classified as a configuration error if:

- A program/utility is installed in the wrong place.
- A program/utility is installed with incorrect setup parameters.
- A secondary storage object or program is installed with incorrect permissions.

For example, at some sites the `tftp` daemon was enabled in such a way that it allowed any user on the Internet to access any file on the machine running `tftp`. This flaw qualifies as a configuration error in our taxonomy because `tftp` was not properly installed. `tftp` should have been enabled such that access to the file system was restricted via the `chroot` command [1, 2].

3.4 Environment Faults

Environment faults are introduced when specifications are translated to code but sufficient attention is not paid to the run-time environment. Environmental faults can also occur when different modules interact in an unanticipated manner. Independently the modules may function according to specifications but an error occurs when they are subjected to a specific set of inputs in a particular configuration environment.

For example, the **exec** system call overlays a new process image over an old one. The new image is constructed from an executable object file or a data file containing commands for an interpreter. When an interpreter file is executed, the arguments specified in the **exec** call are passed to the interpreter. Most interpreters take "`-i`" as an argument to start an interactive shell.

In SunOS version 3.2 and earlier, any user could create an interactive shell by creating a link with the name "`-i`" to a setuid shell script. **exec** passed "`-i`" as an argument to the shell interpreter that started an interactive shell. Both the **exec** system call and the shell interpreter worked according to specifications. The error resulted from an interaction between the shell interpreter and the **exec** call that had not been considered.

4 Selection Criteria

For each of the classifications described in our taxonomy, it should be possible to design a decision process that would help us classify faults automatically and unambigously. Many such decision processes are possible and we present a selection criteria that can be used to classify security faults into different categories to distinctly classify each fault.

For each fault category we present a series of questions that are used to determine membership in a specific category. An affirmative answer to a question in that series qualifies the fault to be classified in the corresponding category.

4.1 Condition Validation Errors

The following sets of questions can be used to determine if a fault can be classified as a condition validation error.

Boundary Condition Errors

- Did the error occur when a process attempted to read or write beyond a valid address boundary?
- Did the error occur when a system resource was exhausted?
- Did the error result from an overflow of a static-sized data structure?

Access Validation Errors

- Did the error occur when a subject invoked an operation on an object outside its access domain?
- Did the error occur as a result of reading or writing to/from a file or device outside a subject's access domain?

Origin Validation Errors

- Did the error result when an object accepted input from an unauthorized subject?
- Did the error result because the system failed to properly or completely authenticate a subject?

Input Validation Errors

- Did the error occur because a program failed to recognize syntactically incorrect input?
- Did the error result when a module accepted extraneous input fields?
- Did the error result when a module did not handle missing input fields?
- Did the error result because of a field-value correlation error?

Failure to Handle Exceptional Conditions

- Did the error manifest itself because the system failed to handle an exceptional condition, generated by a functional module, device, or user input?

4.2 Synchronization Errors

This section presents the criteria that can be used to decide if a fault can be classified as a synchronization error.

Race Condition Errors

- Is the error exploited during a timing window between two operations?

Serialization Errors

- Did the error result from inadequate or improper serialization of operations?

Atomicity Errors

- Did the error occur when partially-modified data structures were observed by another process?
- Did the error occur because the code terminated with data only partially modified as part of some operation that should have been atomic?

4.3 Environment Errors

This section presents a series of questions that be used to decide if a fault can be classified as an environment error.

- Does the error result from an interaction in a specific environment between functionally correct modules?
- Does the error occur only when a program is executed on a specific machine, under a particular configuration?
- Does the error occur because the operational environment is different from what the software was designed for?

4.4 Configuration Errors

The following questions can be used to determine if a fault can be classified as a configuration error.

- Did the error result because a system utility was installed with incorrect setup parameters?
- Did the error occur by exploiting a system utility that was installed in the wrong place?
- Did the error occur because access permissions were incorrectly set on a utility such that it violated the security policy?

5 Applications of Fault Taxonomy

In this section, we present some applications of our fault classification scheme. In addition, we also identified some testing techniques that may be used to systematically detect those faults.

5.1 Vulnerability Database

Landwehr et al.[24] observe that the history of software failure has been mostly undocumented and knowing how systems have failed can help us design better systems that are less prone to failure. The design of a vulnerability database is one step in that direction.

The database could serve as a repository of vulnerability information collected from different sources, could be organized to allow useful queries to be performed on the data, and could provide useful information to system designers in identifying areas of weaknesses in the design, requirements, or implementation of software. The database could also be used to maintain vendor patch information, vendor and response team advisories, and catalog the patches applied in response to those advisories. This information would be helpful to system administrators maintaining legacy systems.

Taimur Aslam designed and built a prototype vulnerability database [8] to explore the usefulness of the classification scheme presented in this paper. Our vulnerability database is based on a relational schema model that consists of both physical and conceptual entities. These entities are represented as relations (tables) in the model. Relational algebra defines the operations that can be performed on the the relations. It also defines a set of basis functions such that any query in the relational model can be specified only in terms of these functions. The basis functions in the relational model are: SELECT, PROJECT, UNION, DIFFERENCE, and CARTESIAN PRODUCT.

The database was populated with vulnerability information from several sources and proved a useful resource in the development of intrusion detection patterns for the COAST intrusion detection system IDIOT [22, 23, 21].

6 Future Work

It needs to be determined whether our classification scheme needs to be enhanced to encompass other operating systems. Many modern systems are based on a software architecture that is different from that of Unix. These include micro-kernels, object-oriented, and distributed operating systems. If needed, our classification scheme can be easily expanded because the criteria used for the taxonomy does not rely on implementation details and is designed to encompass general characteristics of a fault. Also, our existing categories can be extended to include any news faults that cannot be classified into the existing categories, should any be found.

The COAST vulnerability database also needs to be extended with more vulnerabilities. The database currently has over 80 significant faults, largely from variants of the UNIX operating system. We have data to extend the collection to almost 150 cataloged faults. Once this is complete, we intend to evaluate the structure and use of the database for some of our original research goals: building static audit tools, guiding software design and testing, and enhancing incident response capabilities.

7 Conclusion

In this paper we presented a fault classification scheme that helps in the unambiguous classification of security faults that is suitable for data organization and processing. A database of vulnerabilities using this classification was implemented and is being used to aid in the production of tools that detect and prevent computer break-ins. The classification scheme has contributed to the understanding of computer security faults that cause security breaches.

References

[1] CERT advisory CA-91:18. Computer Emergency Response Team Advisory, 1991.

[2] CERT advisory CA-91:19. Computer Emergency Response Team Advisory, 1991.

[3] CERT advisory CA-93:17. Computer Emergency Response Team Advisory, 1993.

[4] CERT advisory CA-94:02. Computer Emergency Response Team Advisory, 1994.

[5] DeMillo R. A, Hocking E. D, and Meritt M. J. A Comparison of Some Reliable Test Data Generation Procedures. Technical report, Georgia Institue of Technology, 1981.

[6] R.P. Abbott et al. Security Analysis and Enhancements of Computer Operating Systems. Technical Report NBSIR 76-1041, Institute for Computer Science and Technology, National Bureau of Standards, 1976.

[7] H. Agrawal, R. DeMillo, R. Hathaway, and et al. Design of Mutant Operators for the C Programming Language. Technical Report SERC-TR-41-P, Software Engineering Research Center, Purdue University, 1989.

[8] Taimur Aslam. A taxonomy of security faults in the unix operating system. Master's thesis, Purdue University, 1995.

[9] Boris Beizer. *Software Testing Techniques*. Electrical Engineering/Computer Science and Engineering Series. Van Nostrand Reinhold, 1983.

[10] Richard Bibsey, Gerald Popek, and Jim Carlstead. Inconsistency of a single data value over time. Technical report, Information Sciences Institute,University of Southern California, December 1975.

[11] Matt Bishop. Analyzing the Security of an Existing Computer System. IEEE Fall Joint Computer Conference, November 1986.

[12] T.A. Budd. *Mutation Analysis of Program Test Data*. PhD thesis, Yale University, May 1980.

[13] Jim Carlstead, Richard Bibsey II, and Gerald Popek. Pattern-directed protection evaluation. Technical report, Information Sciences Institue,University of Southern California, June 1975.

[14] Richard A. DeMillo and Aditya P. Mathur. On the Use of Software Artifacts to Evaluate the Effectiveness of Mutation Analysis for Detecting Errors in Production Software. Technical report, Software Engineering Research Center, Purdue University , SERC-TR-92-P, March 1991.

[15] Daniel Farmer and Eugene H. Spafford. The COPS Security Checker System. Technical Report CSD-TR-993, Software Engineering Research Center, Purdue University, September 1991.

[16] Simson Garfinkel and Eugene Spafford. *Practical Unix and Internet Security*. O'Reily and Associates, 1996.

[17] Katie Hafner and John Markoff. *Cyberpunk: Outlaws and Hackers on the Computer Frontier*. Touchstone, 1992.

[18] W. E. Howden. Reliability of the Path Analysis Testing Strategy. *IEEE Transactions on Software Engineering*, SE-2(3):208–214, 1976.

[19] White L. J and Cohen E. K. A Domain Strategy for Computer Program Testing. *IEEE Transactions on Software Engineering*, 6(3):247–257, May 1980.

[20] D.E. Knuth. The Errors of TEX. *Software Practice and Experience*, 19(7):607–685, 1989.

[21] Sandeep Kumar. *Classification and Detection of Computer Intrusions*. PhD thesis, Purdue University, 1995.

[22] Sandeep Kumar and Eugene Spafford. A Pattern Matching Model for Misuse Intrusion Detection. In *17th National Computer Security Conference*, 1994.

[23] Sandeep Kumar and Eugene H. Spafford. A software architecture to support misuse intrusion detection. Technical Report CSD-TR-95-009, Purdue University, 1995.

[24] Carl Landwher et al. A taxonomy of computer program security flaws. Technical report, Naval Research Laboratory, November 1993.

[25] Brian Marick. A survey of software fault surveys. Technical Report UIUCDCS-R-90-1651, University of Illinois at Urbana-Champaign, December 1990.

[26] G. Myers. *The Art of Software Testing.* Wiley, 1979.

[27] D. Potier, J.L. Albin, R. Ferrol, and A. Bilodeau. Experiments with Computer Software Complexity and Reliability. In *Proceedings of the 6th International Conference on Software Engineering*, pages 94–103. IEEE Press, 1982.

[28] Raymond J. Rubey. Quantitative Aspects of Software Validation. *SIGPLAN Notices*, SE-5(3):276–286, May 1975.

[29] David R. Safford, Douglas Lee Schales, and David K. Hess. The TAMU security package. In Edward Dehart, editor, *Proceedings of the Security IV Conference*, pages 91–118, 1993.

[30] Tsutomu Shimomura and John Markoff. *Takedown.* Hyperion Books, 1996.

[31] Eugene H. Spafford. Extending Mutation Testing to Find Environmental Bugs. *Software Practice and Principle*, 20(2):181–189, Feb 1990.

[32] David M. Weiss and Victor R. Basili. Evaluating Software development by Analysis of Changes: Some Data from the Software Engineering Laboratory. *IEEE Transactions on Software Engineering*, SE-11(2):3–11, February 1985.

PROTECTING COLLABORATION

Gio Wiederhold*
Stanford University, CA

Michel Bilello
Stanford University, CA

Vatsala Sarathy
Oracle Corp., Redwood City, CA

XiaoLei Qian
SRI International, Menlo Park, CA

June 26, 1996

Abstract

The TIHI (Trusted Interoperation of Healthcare Information) project addresses a security issue that arises when some information is being shared among collaborating enterprises, although not all enterprise information is sharable. It assumes that protection exists to prevent intrusion by adversaries through secure transmission and firewalls. The TIHI system design provides a gateway, owned by the enterprise security officer, to mediate queries and responses. The enterprise policy is determined by rules provided to the mediator. We show examples of typical rules. The problem and our solution applies not only to a healthcare setting, but is equally valid among collaborating enterprises and in many military situations.

1 Introduction

We address an issue in the protection of information that is starting to arise as the basic infrastructure for secure transmission and storage enters into practice. We assume an environment where encrypted transmission, firewalls, passwords, and private and public keys provide adequate protection from adversaries. The problem which remains, and addressed here, is now to enable selective sharing of information with collaborators, without the risk of exposing related information in one's enterprise domain or enclave that needs to be protected [1]. We will first sketch some examples to clarify the problem and then formulate the informal model for our work.

In a hospital the medical record system collects a wide variety of information on its patients. Most information on a patient must be accessible to the treating healthcare personnel, including community physicians, and a substantial fraction to the hospital billing clerks [2]. Similar data are requested by insurance companies, and certain data and summarization are due for hospital accreditation and public health monitoring. Results for all of these customers must be handled distinctly.

In a manufacturing company collaborations are often formed with suppliers and marketing organizations. Such virtual enterprises are formed to design, assemble, and market some specific products. Design specifications and market intelligence must be rapidly shared to remain competitive. These collaborations overlap, producing security problems which are stated to be the primary barrier to the acceptance of this approach [3]. Uncontrolled sharing of proprietary data is too risky for a manufacturer to grant a supplier. The supplier will also be wary of giving information to the customers.

In a joint military action situation, information must be shared from a variety of sources with a variety of forces, one's own and allies'. The source information ranges from current force status, logistics backup, to intelligence about the opponents. While opponents should be denied all information, not all of one's troops are authorized to access intelligence sources, and one's allies may be further restricted.

*Supported by NSF grant ECS-94-22688

These three scenarios have the following commonality.

1. We are dealing with friends, not enemies, and should provide relevant information expeditiously.
2. The collected information is not organized according to the needs of a security protocol.
3. It is impossible to rigorously classify the data, a priori, by potential recipient.
4. It cannot be fully determined from the query whether the results combine information which should be withheld.

For instance, a medical record on a cardiac patient can include notations that would reveal a diagnosis of HIV, which should not be widely revealed, and withheld from cardiology researchers. A design document on a plastic component, to be outsourced, also indicates the incorporation of a novel component supplied by another manufacturer, which provides a competitive advantage. Military planning information indicates intelligence sources which are not to be made public to one's allies.

Our model formalizes the role of a security officer who has the responsibility and the authority to assure that no inappropriate information leaves an enterprise domain. A firewall protects the domain vis-a-vis invaders. Distinct gateways, each owned and controlled by a security officer, provide the only legitimate pathways out of, and into, the domain. This gateway is best envisaged as a distinct computer system; we refer to such a system as a "security mediator", placed as sketched in Figure 1. In the security mediator the policies set by the enterprise on security and privacy are implemented, under control of, and through interaction with the security officer. Databases and files within the domain can provide services and meta-data to help the activities of the security mediator, but cannot be fully trusted. The security mediator is able to use secure communication and authentication of outside requests.

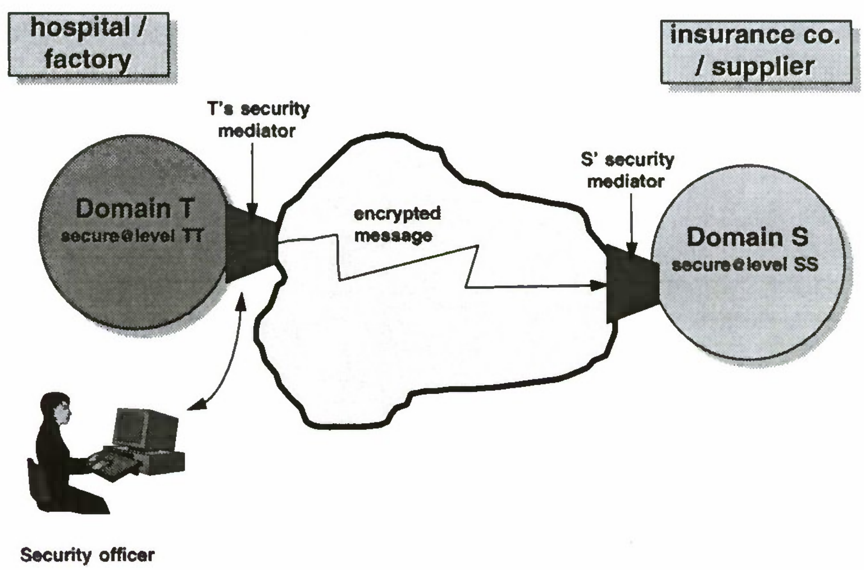


Figure 1: Security mediator setting.

It is important to recognize, as sketched in Figure 2, that validation of communication content must occur both with respect to the query and the responses. For instance, it is inadequate to allow a validated researcher in cardiac diseases to receive all records on cardiac patients, if that also includes HIV cases. Depending on institutional policy, such cases will be omitted or sanitized.

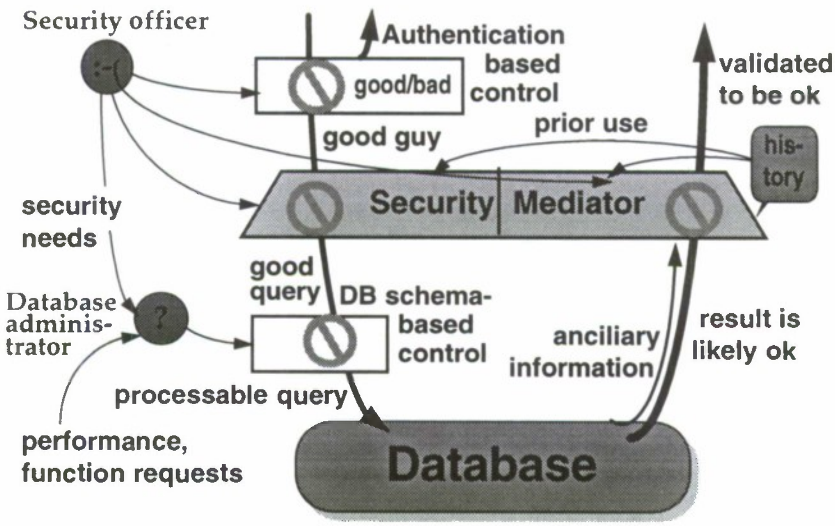


Figure 2: Paths to be checked.

2 System Design

The mediator system consists of modules that performs the following tasks:

- Processing of query (*pre-processing*)
- Communication with databases (submission of query and retrieval of results)
- Processing of results (*post-processing*)
- Writing into a log file

The mediator is designed to safeguard the privacy of the data. There is a two-way fence inside the mediator that intercepts queries coming in and, likewise, results going out. Corresponding to each side of the fence is a set of rules that assesses the legitimacy of queries and results respectively. When a query is sent by a user from the outside world, the mediator applies a set of rules to ensure the query's validity. For example, in a medical application, the mediator will obviously prevent those queries requesting patient names, social security numbers, etc.

The rule system permits fully validated requests and/or validated responses to pass without direct interaction by the security officer, but any other request or response will be presented to the security officer. The security officer then decides whether the request can still be granted. If the results of a query are rejected for rule violation, they are sent to the security officer along with the query and the identity of the user who originated it. If a result should contain information that is questionable, then it is passed to the security officer, who can approve it, edit it prior to approval, or reject it.

The rules balance the need for preserving data privacy and for making data available. Data which is too tightly controlled would be less available and useful for outside users. Conversely, a sufficient level of protection of data privacy must be maintained.

The mediator system can operate fully interactively or partially automatically. A reasonable goal is the automatic processing of say, 90% of queries and 95% responses, but even a fully manual system will provide benefits, as summarized in the conclusions. Even when operating automatically, the security mediator remains under the security officer's control. It does not function like a "black box" but rather keeps the security officer involved in its operation. For example, rules are modifiable by the security officer at all times. In addition, daily logs are accessible to the officer, who can then keep track of the transactions.

The mediator system and the databases typically reside on different machines. Thus, since all queries are processed by the mediator, the database need not be multi-level secure unless it operates in a particularly high security setting.

3 The Rule System

In order to automate the process of controlling access and ensuring the security of information, the security officer must enter rules into the system. The security mediator uses these rules to determine the validity of every query and make valuable decisions pertaining to the dissemination of information. The system helps the security officer enter appropriate rules and update them as the security needs of the organization change.

The rules are simple, short and comprehensive. They are stored in the database with all edit rights restricted to the security officer. If no rules are entered into the database, then the system operates in the manual default mode, whereby access is still possible but all queries and responses pass via the security officer. Some rules may be related to others, in which case the most restrictive rule automatically applies. The rules may pertain to users, sessions, tables or any combinations of these.

Once they are entered into the system by the officer, all the rules will be checked for every query issued by the user in every session. All applicable rules will be enforced for every user and the query will be forwarded only if it passes all tests. Unless a rule permits explicit pass through, it goes to the security officer. In the event a rule is violated by a query, the error message will be directed to the security officer and not to the end user. Thus, in such cases, the users will not see the error message. This is necessary because even error messages could be interpreted and meaningful inferences could be made, or the user could rephrase the query to bypass the error. The errors as well as all queries will be logged by the system for audit purposes.

Because the results retrieved for a given query can be highly unpredictable, it is not sufficient to validate queries. Thus, even when the query has been validated, the results are also subject to screening by a set of rules. As before, all rules are enforced for every user and the results are accessible only if they pass all tests. Also, if the results violate a rule, an error message is sent to the security officer but not to the user.

Not only are the rules easy to comprehend and to enter into the system, they are also powerful enough to enable the officer to specify requirements and criteria accurately, so that whenever users may see all information, they should be allowed to do so and whenever information is restricted, they should not have access to it. The users in the system are grouped as cliques and rules may apply to one or more cliques. The security officer has the authority to add or delete users from cliques and to create or drop cliques. Similarly, columns in tables can be grouped into segments and query/results validations could be performed on segments.

The rules can be classified as *set-up* or maintenance rules, *pre-processing* (query) rules and *post-processing* (result) rules. Some rules may be both *pre-* and *post-processing* rules. Examples of *pre-processing* rules include number of queries per session for the clique, session time, session hours, statistical query only, etc. *Post-processing* rules include minimum rows retrieved, session time, intersection of queries, user hours, vocabulary matching. A more comprehensive list of rules can be found in appendix. The rule type is indicated in parenthesis.

3.1 Application of rules

The following sequence of rules is applied for every request.

- When the user enters a query, the mediator parses the query. If parsing is not successful, an error message is sent out to the security officer.
- Next, the security mediator checks to see if the user belongs to a clique. If not, an error message is sent to the security officer.
- Then, it checks to see if access to all the columns specified in the SELECT and WHERE clauses in any segment is permitted to the members of the clique. If not, an error message is sent to the security officer.
- It then looks at every rule in the system of type *pre-processing* and validates the query against each. If any rule is violated, an error message is sent to the security officer.

At this point, the query is actually processed and results are obtained by the mediator.

- Now the *post-processing* rules are applied.
- On textual results, rules may specify that all words must come from a specified vocabulary. Any unknown term will be presented, with surrounding context, to the security officer, and if not approved, no result will be returned.
- Security officers can edit documents brought to their attention before releasing them. That should include 'whiteing-out' portion of graphics and design drawings.
- Lastly, further result modification is done as specified by the rules. Operations that can be invoked include random falsification of data and aggregation.
- Now the results are sent back to the user. Then the mediator updates internal statistics such are number of queries for the session, duration of the session, etc. It also updates the log files appropriately. This last step is done in all cases, whether or not there were errors.

4 View-Based Access Control

Most databases in place today were originally developed for internal use only. The security mechanisms available in these systems are intended for access by only a known, controllable, observable, and predominantly loyal internal user population, rather than unknown, unseen, and potentially adversarial external user populations [4]. Consequently, while internal access control based on user discretion might be satisfactory, external access control should support mandatory enforcement, before an enterprise can comfortably share its data with other partners in a collaboration.

Notice that the tables referred to in rules do not have to be base relations. They can be derived relations or views defined by arbitrary SQL queries. Hence, the set of rules collectively specifies a view-based access control policy.

Views in relational databases have long been considered ideal as the objects of access control, because they have a higher degree of logical abstraction than physical data and hence enable content-based or context-based security, as opposed to container-based security provided in operating systems.

View-based access control in relational databases was first introduced in IBM's System R [5], in which views expressed in SQL are the objects of authorization. It has been adopted by most commercial relational DBMSs. However, view-based mandatory access control has not been in widespread use because of the safety problem [6]. The safety question asks the following. Is there a database state in which a particular user possesses a particular privilege for data in a specific view? In container-based access control, different containers do not share contents. Hence, a secret label on a container guarantees that data in the container are not accessible to unclassified users. In view-based access control however, views might overlap because the same data might satisfy more than one view. Hence, a secret label on a view does not guarantee that data contained in the view are not accessible to unclassified users.

To support view-based mandatory access control, queries have to be analyzed and answers have to be filtered to ensure that data in a view are accessible by all and only those users who are authorized to access the view. We envision two types of query analysis.

1. *Analysis of single queries.* A query should be sufficiently constrained such that it only accesses those views to which the issuer of the query has authorization.

2. *Analysis of a sequence of queries.* A sequence of queries by the same issuer should be sufficiently constrained such that the issuer cannot compute, from the sequence of answers, data in views to which he does not have authorization.

4.1 Single Queries

The easiest way of enforcing mandatory access control is of course to require that a query be formulated in terms of those views to which the issuer of the query has authorization. For example, suppose that the following view is defined:

```
CREATE VIEW Drug_Allergy (patient_name, drug_name, notes)
SELECT      Patients.name, Drugs.name, Allergy.text
FROM        Patients, Drugs, Allergy
WHERE       Patients.id = Allergy.patient_id
AND         Drugs.id = Allergy.drug_id
```

on which the following rules are specified:

```
CREATE CLIQUE X
ADD USER      John_Doe X
LIMIT         X Drug_Allergy.
```

Then queries issued by user John Doe have to be formulated in terms of the view Drug_Allergy. For example, the following query by John Doe will be rejected by the security mediator,

```
SELECT Patients.name, Allergy.text
FROM   Patients, Drugs, Allergy
WHERE  Patients.id = Allergy.patient_id
AND    Drugs.id = Allergy.drug_id
       Drugs.name = xd_2001
```

even though it is equivalent to the following query, which will be accepted by the security mediator.

```
SELECT patient_name, notes
FROM   Drug_Allergy
WHERE  drug_name = xd_2001.
```

Therefore, the security mediator should not base acceptance decision of a query on the condition that the issuer of the query has authorization to all relations mentioned in the query, base or derived. Instead, the security mediator should try to reformulate the query using those views that the issuer of the query has authorization. If a reformulation is possible, then the reformulated query will be evaluated in place of the original query. Otherwise the original query is rejected. This approach will also facilitate the evolution of the security policy enforced by the security mediator.

4.2 Sequence of Queries

Access control on a per-query basis might not be sufficient. Even when a user has authorization to every query issued, he might be able to combine answers from a sequence of queries to derive data in a view to which he does not have access authorization. Such scenarios necessitate the need for the security mediator to keep track of the access history for every clique/user. For example, even if user John Doe is not authorized to access the view Drug_Allergy, he could issue the following two queries, assuming that he is authorized to both, and obtain data contained in the view Drug_Allergy.

```
SELECT Allergy.patient_id, Allergy.drug_id, Patients.name, Allergy.text
FROM   Patients, Allergy
WHERE  Patients.id = Allergy.patient_id.
```

```
SELECT Allergy.patient_id, Allergy.drug_id, Drugs.name, Allergy.text
FROM   Drugs, Allergy
WHERE  Drugs.id = Allergy.drug_id.
```

A critical issue in analyzing a sequence of queries is what we can assume about the computational capability of the user in combining the sequence of answers. For the above example, John Doe has to be able to perform join over the answers of the two queries in order to compromise the view Drug_Allergy. A reasonable assumption is that users have the same computational capability as in single queries. In other words, if users can issue project-select-join queries, then they can perform project, select, and join operations on a sequence of answers.

Another important problem is when queries are interleaved with updates, because even though John Doe might have already accessed a portion of the data in the view Drug_Allergy, say the first query above, enough time might have elapsed before he issues the second query above that the join between the two answers is empty. This could happen if for example the base relation Allergy only contains data for the most recent month, and John Doe waited over a month to ask the second query. In this case, the history log for queries on relation Allergy could safely be bound to one month.

Therefore, the security mediator should try to reformulate the view Drug_Allergy that John Doe is not authorized to using queries issued by John Doe. If a reformulation is possible, then the security policy on Drug_Allergy is violated.

5 Conclusion

We are addressing privacy and security maintenance in collaborative settings, where information has to be selectively protected from colleagues, rather than withheld from enemies. The problem only arises once a basic secure infrastructure is established. Today, privacy protection in healthcare is preached, but ignored in practice, putting many institutions at risk. In crucial settings, corporate and military security officers control input and output, but do so on paper, so that interactions are typically delayed by weeks, and high costs are incurred due to delays and misunderstandings. The primary barrier, as stated in [3], to the realization of virtual enterprises is 'Insufficient security controls. The corporations participating in a virtual enterprise are independent and frequently compete against one another'.

✪ Be helpful to customer	**➲ Be helpful to security officer**
✪ Tell cust. re problems, ***query may be fixed***	**➲ Tell cust. re problems,** ***sec. off may contact cust.***
✪ Exploit DB meta-data	**➲ Exploit customer info.**
✪ Isolate transactions	**➲ Use history of usage**
✪ Ship result to customer	**➲ Ship result to sec. off. with result description (source, cardinality)**

Figure 3: Differences in mediation for queries and for protection.

The approach we are developing provides tools for a security officer. Database systems have provided tools to control queries, under the aegis of the database administrator. We illustrated above that query-only tools are inadequate in complex settings, and we emphasized the need for view-based access control. In addition, the major role of a database administrator is to help customers get maximal relevant data, a task that often conflicts with security concerns as illustrated in Figure 3. Furthermore, the majority of data is not in database systems that provide security, and even less resides in costly, validated multi-level secure systems.

The concept of security mediator as an intelligent gateway protecting a well-defined domain is clear, simple, and the cost of modern workstations make it feasible to assign such a tool to a security officer. Like most security measures, the security mediator cannot offer a 100% guarantee, especially with respect to statistical data security. But having a focused node, with a complete log of requests and responses, and an incrementally improving rule collection, provides a means to ratchet protection to a level that serves the enterprise needs and policies effectively.

The authors wish to thank Dr. Lee Mann of Inova Health System for valuable discussions and for providing test data.

Examples of Rules

Rule	Remarks
1. `set logfile "x"` (Set up)	The table or path name to the log file
2. `create clique x` (Set up)	Create a clique of users called x
3. `add user user_name clique_name` (Set up)	add user called user_name to clique_name
4. `delete user user_name clique_name` (Set up)	
5. `drop clique x` (Set up)	
6. `create segment segment_name` (Set up)	
7. `set stat_only true/false` (Pre)	Only statistical info (average, median) allowed
8. `set clique stat_only true/false` (Pre)	Only statistical info (average, median) allowed for user
9. `set segment stat_only true/false` (Pre)	Only statistical info (average, median) allowed for queries on given table
10. `set user table stat_only true/false` (Pre)	Only statistical info (average, median) allowed for user, table combination
11. `limit queries_per_session x` (Pre)	Number of queries allowed in a session
12. `limit clique queries x` (Pre)	For a given user, number of queries allowed per session
13. `limit clique segment` (Pre)	limit all users in clique to columns/tables in segment. This specifies explicit pass through of results.
14. `set random on/off` (Post)	Random falsification of data to be performed or not
15. `set random on/off clique` (Post)	Random falsification of data to be performed or not for user
16. `set random on/off segment` (Post)	Random falsification of data to be performed or not for queries on given table
17. `set user table random on/off` (Post)	Random falsification of data to be performed or not for user/table combination
18. `limit min_rows_retrieved x` (Post)	Minimum number of matching rows for a given selection criterion
19. `limit clique min_rows x` (Post)	Minimum rows retrieved for a query by a given user
20. `limit segment num_queries x` (Post)	Number of queries allowed on a given table
21. `limit clique segment num_queries x` (Post)	Number of queries allowed on a given table for a given user
22. `limit intersection x` (Post)	No two queries can have an intersection greater than x rows
23. `limit clique intersection x` (Post)	No two queries by user can have an intersection greater than x rows
24. `limit segment intersection x` (Post)	No two queries on table can have an intersection greater than x rows

References

[1] D. Randolph Johnson, Fay F. Sayjdari, and John P. Van Tassell. Missi security policy: A formal approach. Technical Report R2SPO-TR001-95, National Security Agency Central Service, July 1995.

[2] Bill Braithwaite. National health information privacy bill generates heat at scamc. *Journal of the American Informatics Association*, 3(1):95–96, Jan/Feb 1996.

[3] Martin Hardwick, David L. Spooner, Tom Rando, and KC Morris. Sharing manufacturing information in virtual enterprises. *Comm. ACM*, 39(2):46–54, February 1996.

[4] G. Rettig. Use of multi-level secure systems in commercial environments. January 1991.

[5] P. P. Griffiths and B. W. Wade. An authorization mechanism for a relational database system. *ACM Transactions on Database Systems*, 1(3):242–255, September 1976.

[6] M. Schaefer and G. Smith. Assured discretionary access control for trusted RDBMS. In *Proceedings of the Ninth IFIP WG 11.3 Working Conference on Database Security*, pages 275–289, 1995.

DESIGN AND MANAGEMENT OF A SECURE NETWORKED ADMINISTRATION SYSTEM : A PRACTICAL APPROACH

Vijay Varadharajan
Professor of Computing, University of W. Sydney, Australia.
email : vijay@st.nepean.uws.edu.au

June 7, 1996

Abstract

As applications become more distributed, the design and management of security services in networked systems play an increasingly significant role. This paper describes the design of services for securing the management of a networked administration system. It presents the architectural principles involved and the overall security solution comprising the design of security services and the trusted components that provide these services. The security schemes are illustrated by providing a walkthrough of the networked administration scenario.

1. Introduction

Security plays a vital role in the design, development and practical use of the distributed computing environment, for greater availability and access to information in turn imply that distributed systems are more prone to attacks. The need for practical solutions for secure networked system management is becoming increasingly significant. In developing these solutions, several important issues need to be carefully addressed. The design of the required security services forms a major part. Often the issues associated with security management are not adequately addressed. First, it is important to identify clearly the functionalities and interfaces of the trusted security management components. Then it is necessary to consider whether some of these trusted management authorities can be grouped together to simplify the overall management. This depends on several factors such as the relationships between organizations (or units) involved in the networked environment and the types of services offered as well as performance considerations. In practice, it is also necessary to consider the system development and deployment in stages thereby enabling a staged adoption.

In this paper, we address the design and management of a secure networked administration system. The paper is organized as follows: Section 2 describes a network administration scenario, and outlines the different stages involved in the development of the system. Section 3 discusses the architectural issues and outlines the design of security services and the provision of security facilities. The secure system operation is described in Section 4. We outline the different phases involved in the life of a user, application and the system, and describe how the security services are managed by the various components in the architecture. Finally Section 5 provides a walkthrough of the network administration scenario and illustrates the use of security services and facilities.

2. Secure Networked Administration System Design

2.1 Scenario

The scenario we consider is an example demonstration of a secure distributed application. The scenario involves administration of multiple hosts in a network using a single administration station from which authorized security managers can perform various administration functions (See Figure 1). The application that we consider is a distributed configuration and auditing of networked systems. It involves such tasks as configuration of audit scripts (for instance, specifying what checks to be done), collection of audit information, and browsing the audit data.

In a large practical networked environment, there will be several managers responsible for different parts of the network. Our scenario allows different security managers to have different sets of privileges. For instance, Security Manager A might be responsible for hosts 1,2, 3 and 4, and might have authority to configure, audit and browse audit information of hosts 1,2 and 3, and only browse audit information of host 4, whereas Security Manager B is responsible for hosts 3 and 4, and has authority to configure, audit and browse host 4 and only has browse authority for host 3. More generally the privileges capture both geographical partitioning of the networked hosts as well as the type of actions that a manager can perform over the hosts.

To ensure that only authorized entities are able to set the configurations and control the audit process, it is necessary to provide mutual authentication between the security administration agents and the remote hosts. Furthermore, secure transfer of information between remote hosts and the security administrator workstations is required. Hence this scenario brings together issues of privilege control, authentication, secure communication and auditing in an integrated manner.

In addition, the task of administering networked systems is a round the clock activity. Hence it may be necessary for the security manager to access the security administration workstation remotely, e.g. from home or from a different location in the site. For instance, the manager may browse through the security status of the network system before determining whether a visit to the site is required. However the set of privileges that a manager has while accessing from a remote location is likely to be different from those that she has while physically present at the administration workstation[1]. Our scenario envisages secure remote access using a mobile personal information appliance such a palmtop computer over either a public switched telephone network or a wireless network.

The major stages of the Secure Networked Administration System (SNAS) development are (See Figure 2) :

(a) from a single Security Administration Station (SAS) with a single Security Manager.

(b) from a single SAS with mutiple Security Managers responsible for different parts of the network system, and having different sets of privileges.

(c) from a single SAS with mutiple Security Managers, with remote access to SAS from a mobile device (dial in/wireless).

(d) with mutiple SAS - one SAS per domain. (A domain comprises a collection of hosts over which a single SAS has jurisdiction).

3. Secure System Design : Architectural Issues

3.1 Design Goals

The basic set of design goals, related both to the definition of the services provided by the components and their implementation are as follows :

- With respect to the development of such a secure system, the aim is not to produce a monolith. We consider this to be in phases thereby enabling a staged adoption.

[1]For instance, this could be a proper subset.

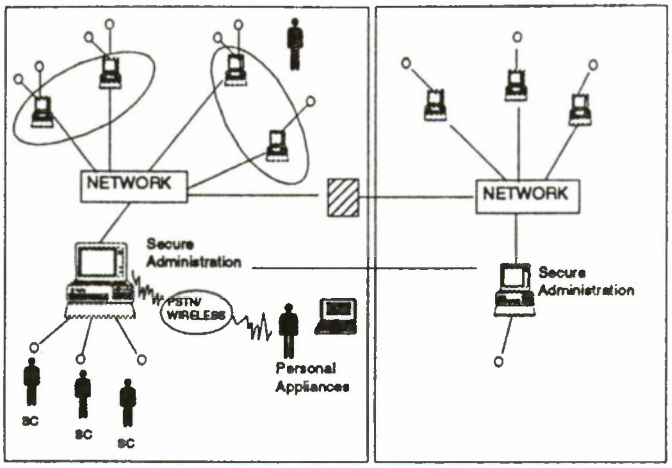


Figure 1: Network Administration Scenario

- Uniform treatment of agents acting as principals, no matter what kind of agents they are (person, hardware or software component)
- The implementation of components will be heavily dependent on the operating system interfaces. However the model of the operating system that we have assumed applies to a broad range of hosts, allowing re-implementation of the same service definitions and protocols as necessary.
- The choice of cryptographic algorithms is an important one due to licensing and export control issues, as well as technological feasibility. This is not a question of providing many protocols, but of implementing them behind uniform service definitions, so that the application developer can work independently of this decision.
- Support management of security information wherever it is distributed, not just at a central location. Also, the aim is to bring the choice of mechanisms behind the service definitions into the management world, not forcing the application developer to hardwire them.
- Integration of security management with network and system management, thereby providing a uniform management view to the administrator.

3.2 SNAS Services and Facilities

The security services provided by SNAS are the following:

- *Authentication Service* : This service supports authentication of both interactive (e.g. a human user) and non-interactive principals (e.g. applications) [7].
- *Authorization Service* : This service allows an application to decide whether a request for a particular service by another principal is to be granted or not [8].
- *Secure Communication Service* : This service provides secure communication of information transferred between remote principals. Secure communication here implies confidentiality, integrity or both.
- *Auditing Service* : The auditing service considered in SNAS provides a snapshot of the system at a given time, thereby allowing a security administrator to easily inspect the security status of the system.

Our approach is based on a hybrid technology using both public key as well as symmetric key systems. One may view this approach as an extension of the Kerberos [3] and DCE [4] systems, which are at present based on symmetric key technologies. The DCE is planning to introduce the public key technology in an incremental manner. A version of public key based Kerberos has also appeared in [12].

We also introduce the concept of an Authorization Server which captures more sophisticated access control information compared to the Privilege Server in the DCE (which primarily deals with groups). The access control information that we consider have different static and dynamic characteristics. Role is an example of such access control information. More significantly, the architectural as well as the design issues described in this paper should be relevant to future DCE extensions.

We now describe the design and operation of these services by considering

- the trusted components of the architecture that are involved in the provision of these services,
- the security information and attributes used by these services and where they are stored and how they are distributed, and
- the different phases involved in the life of a user, application and the system.

3.3 Principals

Principals are the basic elements over which access control can be exercised. A principal is the smallest entity that can be authenticated across a collection of machines in a domain. Thus, for a domain comprising Unix machines, a principal is a map from machines to UIDs.

Let us now consider the trusted principals that exist in our architecture.

We have a single *Certification Server* (CS) principal, which is a global entity in a domain, and an *Authentication Server Component* (ASC) principal on each machine. The CS retains keys associated with the principals and the ASCs. To avoid the need to securely install the key of every principal in the database of every other principal, the CS has been provided.

We have a *Rolebase Server* (RS) principal. For the moment, we will assume one such entity per domain, though there is no reason why there should not be several such entities. The RS has information on which users (principals) have what roles in the domain. E.g. a user Fred is a accountant in organization X. This role information is assumed to be of a general type. We have an *Authorization Broker* (AB) principal on each machine. AB performs the following functions. First, it provides an application principal in a machine the role of a user who is binding to the application. AB obtains this information by contacting the RS. Secondly the AB at the target end verifies the authenticity of the role information provided by the client. Thirdly, at the target end, AB checks the access control information (ACI) – which privileges what users (based on ids and roles) have –, and advises the target application on whether to grant the request or not. The ACI is stored at the target.

Hence we have the following trusted principals (See Figure 2):

- For Authentication Service
  - Authentication Server Component (ASC)
  - Certification Server (CS)
- For Authorization Service
  - Authorization Broker (AB)
  - Rolebase Server (RS)

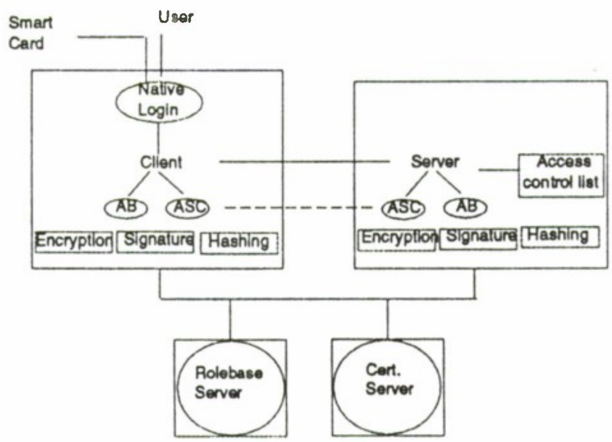


Figure 2: Security Components and Trusted Authorities

3.4 Security Information in SNAS

There are two types of security information involved in these various phases in SNAS, namely that are stored in various security components and that are transferred between components.

Let us consider the characteristics of the different types of security information. Some security information are of *generic* and *static* in nature. Identity based authentication information typically falls into this category. Some security information are *specific* and still somewhat *static* in nature. Role based information falls into this category. Roles are specific to organizations and they are reasonably static in the sense that they are unlikely to change on a day to day or even on a monthly (or even yearly) basis. In fact, one of the main benefits of the role based access control is to reduce the effect of the changes in the user population on the management of access privileges. Then we have security information that are *specific* and *dynamic* in nature. Specific in the sense that they may relate to applications and/or parts of applications such as files. They tend to be dynamic in the sense they are prone to changes as and when updates are made to applications and functionality changes occur.

Furthermore, the authorities involved in the management of these different types of security information are likely to be different. Not only the strategies with respect to *when* the changes and updates to these information take place are likely to be different (mentioned above) but also *who* are allowed to make these changes are likely to vary. For instance, the specification and changes to the role information in an orgnization will be the responsiblity of a certain group of people who can be different to those responsible for setting the privileges for a specific file or application in a server.

3.5 Design Principles

From an architectural point of view, such a characterization leads to the following design principles [8].

Principle 1

Store the static and generic information in some form of a central server responsible for a collection of clients and servers (in a domain).

Principle 2

Store the dynamic and specific information near or in the end system where the target applications reside, enabling the end system authorities to be involved in their management.

The above characterization also affects the way the security information is being distributed.

Principle 3

Static and generic information, being stored in a central server in a domain, can be "pushed" by the client to the target application server. In fact, static and specific information can also be "pushed" in a similar fashion.

Principle 4

Specific and dynamic information needs to be "pulled" at the time of the decision process.

It is important to note that these two types of information may be stored in two different servers owned and managed by different authorities. Based on these principles, one can certainly argue for the need for two trusted authorities —- one dealing with generic and static security information and the other dealing with specific and static security information —- both of which can be architected as central servers servicing multiple clients and servers within a domain. These two correspond to the Certification Server and the Rolebase Server in our architecture. The Certification Server stores the authentication certificates of the principals, which are static and generic. The Rolebase Server stores the user identities and the roles (and their generic privileges) that can be taken by these identities. These are organization-specific and are still relatively static.

The target server stores specific and dynamic security information; often such information are dependent on the state of the application or resource under consideration. Such information include attributes associated with specific rights in the application. For instance, the client might be allowed to withdraw 10000 dollars from Monday to Friday. He might have withdrawn 4000 dollars on Monday, leaving her only 6000 dollars for the rest of the period. So when the client makes subsequent requests, the previous state associated with the transaction needs to be taken into account.

The system's security information is captured using the following constructs:

- A Certificate containing the identity and the public key related information transferred from the Certification Server to the Authentication Server Component. This is signed using the private key of the CS.
- An Authentication Token between the client and the server ASCs for mutual authentication. This is protected using the public key of the target (or client) and signed using the private key of the client (target).
- A Token containing the identity and role information transferred from the Rolebase Server to the Authorization Broker. This token is signed by the Rolebase Server using its private key.
- Access Control Information representing the dynamic and specific information and state dependent information residing at the target end systems.
- Secure conversation between client and target principals, protected using symmetric conversation key established at the end of the mutual authentication process.

3.6 Authorization Service Design

The design of authorization service for distributed applications is an important topic and it merits a separate paper of its own which is in preparation [8]. Here we outline some of the relevant features that form part of the Rolebase Server and the Authorization Broker in SNAS.

The administrator of a networked system in an orgnaization needs to manage privileges of individuals in terms of group profiles, department membership and so on. Furthermore the "give" rights of various administrators need to be configured. Hence the need for a policy language. The policies expressed in this language must be translated into a form usable by the Authorization Broker at access decision time. In particular, the representation of the policies at administration time at the Rolebase Server is likely to be different from the representation of the policy at runtime used by the Authorization Broker.

The syntax and semantics of the language is described in [8]. Here we just outline the logical components of the Rolebase Server:

- An Administration Store : Stores the policy expressions whose interface allows an administration client (user) to input and modify policy statements.
- An Evaluation Store : Stores the policies expressions in a representation suitable for runtime access and decision. As mentioned above, this is different from the administration policy representation in that here one can compile out the semantics of inheritance and overrides in the expressions, thereby making the access decision faster, for instance, by avoiding the need to search the inheritance hierarchy.
- A compiler that translates the policy expressions from the administration time representation to evaluation time representation.
- An engine that evaluates and services a query, and encapsulates the privilges in the form of an Authorization Token and passes it to the requesting client. The Token is passed to the Authorization Broker of the Server which interprets and evaluates the authorization infomation along with its locally stored ACI to make the access decision.

4. System Operation

We present the characteristics of the system by outlining the operations involved in the different system phases.

4.1 Phases

We identify the following phases in the system.

4.1.1 Installation Phase

In the **Installation phase**, we assume that all the required software components of SNAS are correctly installed. We will assume that the Rolebase is also initialized. We will also assume that the access control lists and the mapping from roles to privileges at the (target) servers have also been initialized.

4.1.2 Certification Phase

In the **Certification** phase, the principals are identified to the CS and the keys associated with them are registered with the CS. In the case of machine principals, the keys are public keys, and the CS creates certificates. A certificate comprises the name, the Id, the public key of the principal, and a validity period, signed by the CS's private key. Hence CS stores certificates of ASCs of different machines (including Rolebase Server). We assume that the public keys of the Certification Server and the Rolebase Server are known to all ASCs in the system. For users with smartcards, we can store the private keys in the smart card. If the smart card technology only allows symmetric key based computation then we have the secret symmetric key of the user stored securely in the smartcard and in the CS.

4.1.3 Booting Phase

In the **Booting** phase, when a machine is switched on, the ASC of that machine authenticates itself to the CS using a challenge-response protocol. The CS sends a challenge to which the ASC produces a response using his private key of the public key system. Recall that the ASC has registered its public key with the CS during the certification phase. Following a successful challenge-response protocol, a connection number is established between the ASC in the machine and the CS, which is subsequently used when a principal (user or an application) in that machine requires information from the CS.

4.1.4 Session, Binding, Request and Message Phases

Consider the situation where a user U wishes to log on to a machine X, and an application A_x in machine X acting on behalf of user U invokes an application B_y in machine Y for a service. A_x is acting as a client and B_y is acting as a server.

Let us first consider the **Session** phase. In this phase, an agent acting as a principal presents itself to the system : in effect, to the CS. In the case of users, this process involves a login facility and may involve the smartcard, if this is being used.

Following the certification phase, recall that both the public key of the AS in machine X and the secret symmetric key of the user smartcard have been registered with the CS.

The challenge-response protocol to establish the initial user authentication as follows :

The user logs on by providing his Id and his PIN. The login facility passes this information to the smartcard (SC) which checks the validity of the PIN. The use of the login initiates a session with the ASC on the machine. The ASC now sends the principal Id to the CS, signed by the private key of the ASC. The CS generates a fresh nonce as a challenge and the corresponding response using the user secret symmetric key. The challenge-response pair is then signed using the private key of the CS and sent to the ASC. Now the ASC passes the challenge to the SC (via the login facility). The SC calculates the response and sends this to the login which is then able to verify by matching it with the one received from the CS.

When a principal in machine X (e.g. A_x) wishes to request a service from another principal (e.g. B_y) on the remote machine Y, their respective ASCs will need to communicate. If it is the case that the ASC of machine X is not aware of the ASC of machine Y, then it will make use of the CS Certification Server as a directory to obtain the certificate containing the public key of B's ASC. (Once an ASC has obtained a certificate, this can be cached.) Now using the public key of Y's ASC, X's ASC can establish a conversation key which is used in the protection of communications between the principals A and B. This phase is referred to as the **Binding** phase, which concludes with the establishment of a secure channel between the client (A_x) and the server (B_y).

Then comes the **Request** phase where A_x makes the request for a service to B_y using the established secure channel. Before this happens, the client A_x talks to the Authorization Broker (AB) principal to find out the role of the user who is binding to it. It provides AB the authenticated Id of the user. AB then has a conversation with the Rolebase Server machine. Note that this conversation needs only to be protected for integrity and authenticity and not for confidentiality. This is because the user to role mapping is not likely to be sensitive information. The Role Token captures the user Id, the Role information and its associated privileges along with timestamps. This information needs to be verified by the AB of the target server. These requirements are met using a public key based protocol between the AB and the RS. Recall that following the certification phase, the public key of RS is known to AB.

In the **Message phase**, peer to peer communication between the principals A and B occur. These messages are protected using the conversation key established above. Note that we have a different conversation key whenever a new binding between two ASCs occur. For instance, if two principals A_x and B_y complete one conversation and then have a second conversation, then the conversation key would be different in the two cases. Protection here could be just confidentiality (using encryption), or integrity (using cryptographic checksums), or both. The ASC and CS are not involved. The secure communications facility allows the application programmer to set up such connections and use the agreed algorithms and keys transparently, as a secure version of TCP.

B_y now has to decide whether or not to grant the request from A_x. B_y requests the AB to verify the claimed role of the user who is making the request via A_x. AB communicates with the access control information (ACI) database, which contains information on what privileges are allowed for what user identities and roles, and for what applications. At present, we assume that this ACI resides locally on the target machines. AB interprets this information and advises the application on whether to grant the request or not.

A full description of the security protocols involved in the different phases above can be found in [9].

5. Example Walkthrough

Let us now return to the network administration example.

We have a Security Administration (SAS) station which runs management applications for configuring, collecting, analysing and presenting audit information in a networked system. It provides secure administration of network of Unix systems from a single management station by authorized users - security and network system administrators. From this central station, the security administrator can easily evaluate the level of security at remote systems. It provides quick inspection of security status of the networked system and helps to maintain a minimum level of security. In particular, it is intended to provide snapshots of the system at chosen times to point out existing security anomalies (cf. health checks).

There are audit agent applications residing in each of the remote system that needs to be administered. User U logs onto the secure administration (SAS) workstation, and invokes an audit management application A. The audit management application, acting on behalf of U, requests service from a remote audit agent application B residing in one of the hosts to be administered. We will refer to this host as Y. The request could involve configuration of audit files and audit checks in remote audit agents, activation of audit agents, and transfer of information pertaining to the security status of the remote host and related audit data.

The user U with the smart card is first authenticated using the ASC of the SAS and the Certification Server. The ASCs of the SAS and Y communicate to mutually authenticate each other and establish a common conversation key. This establishes a secure channel between the audit management application A and the remote audit agent B.

The next step is to establish the privileges of the user in question, using the procedures described above. This involves the Authorization Broker of the SAS communicating with the Rolebase Server to determine the role and privileges of the user U. This is used to establish the fact that the user U can have an administrator role and determine the generic privileges associated with this role. The signed role information and the certified identity information (obtained from the Certification Server) are passed to the remote audit agent B, along with the request. The relevant parts of the communication are protected using the previously established conversation key between A and B. B now requests its Authorization Broker (AB) to verify the claimed role of the user making the request and determine the access rights using the Access Control Information (ACI) database. AB interprets this information and advises the audit agent application B on whether to service the request from A.

5.1 Specific Implementation Choices

In this particular application, the SAS performs the administration functions for a networked system of clients and servers. Given this role for the SAS, it is natural for it to hold the role and access privilege information. That is, an implementation choice is made to co-locate the Rolebase Server with the SAS. Note that from the design point of view, the interfaces of the Rolebase Server remain the same. However with this implementation it is not necessary to protect the communication channel between the AB and the RS as they occur within the system.

Stage (b) of SNAS specifies privileges of the various Security Managers in its Rolebase Server. The privilege expressions capture both the range of hosts and subnets that are to be managed by a Security Manager as well as the classes of actions that the Manager has the authority to perform. For instance,

- Manager A can perform actions *Audit Hosts 1,2 and 3 in Subnet N1* AND *Configure Hosts 2,3 and 4 in Subnet N2*, AND *Audit all Hosts in Subnet N2.*
- Manager B can *Configure and Audit all Hosts in Subnet N1.*

The language used to specify the privileges, and their representation and management is described in [8].

Stage(c) involves two additional aspects. The first aspect is the authentication of the remote user and the mobile device over a wireless or dial-in connection. Challenge-response technique similar to the one described earlier has been used to achieve this. Hence we will not describe this here. The second aspect concerns the difference in privileges of a Security Manager when accessing the SAS remotely over a wireless network using a mobile appliance compared to the same Manager accessing the SAS while physically present at the administration workstation. This difference in privileges is captured as part of the policy specifications in the Rolebase Server residing within the SAS. Once again, the language constructs have been designed in such a way to cater for these situations.

Regarding the cryptoalgorithms, appropriate choices are 512-bit RSA for public key based authentication, DES for encryption of data communications, and MD5 for generating hashed message digests.

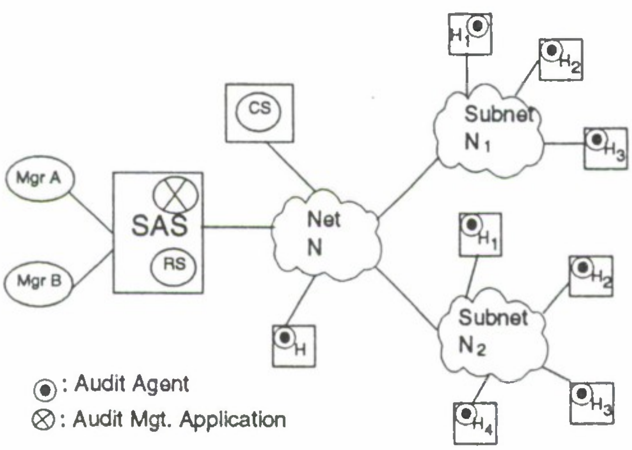


Figure 3: Secure Network Administration System

Acknowledgements : The author would like to thank the anonymous referees for their valuable comments.

References

[1] Y.Yemini, "Emerging Trends in Networks and System Management", Third International Symposium on Integrated Network Management, San Francisco, USA, 1993.

[2] M.Gasser et al, "The Digital Distributed System Security Architecture", National Computer Security Conference, Baltimore, USA, 1989.

[3] Clifford Neumann, Ts Theodore, Kerberos : An Authentication Service for Computer networks, IEEE Communications, Vol.32, No.9, Sept.1994.

[4] Open Software Foundation (OSF), DCE Security Services, Vers.1.1, 1995.

[5] J.J.Tardo, K.Alagappan: "SPX: Global Authentication using Public Key Certificates", Proc. of the IEEE Conference on Security and Privacy, 1991.

[6] V.Varadharajan, P.Allen, S.Black, "An Analysis of the Proxy Problem in Distributed Systems", Proc. of the 1991 IEEE Symposium on Research in Security and Privacy, 1991.

[7] Lampson et al, "Authentication in Distributed Systems : Theory and Practice", ACM Trans. on Computing Systems, Vol.10, No.4, 1992.

[8] V.Varadharajan, "Authorization in Distributed Systems", In preparation. Abstract was presented at 1995 IEEE Symposium on Security and Privacy, Oakland, as a Short Presentation.

[9] V.Varadharajan, "Design of a Secure Network Administration System", Technical Report, UWS Computing, 1995.

[10] V.D.Gligor, S.W.Luan, J.N.Pato, "Om Inter-Realm Authentication in Large Distributed Systems", Proceedings of the IEEE Conference on Security asnd Privacy, 1992.

[11] R.Yahalom, B.Klein, T.Beth, "Trust based Navigation in Distributed Systems", Computing Systems, Vol.7, No.1, 1994.

[12] R.Ganesan, "Augmenting Kerberos with Public Key Cryptography", ISOC Symposium on Network and Distributed System Security, 1995.

Information Warfare, INFOSEC, and Dynamic Information Defense*

J.R. Winkler, C.J. O'Shea, M.C. Stokrp

PRC Inc.
1500 PRC Drive
McLean, VA 22102
(703) 556-1000
winkler_vic@prc.com oshea_connie@prc.com stokrp_mark@prc.com

ABSTRACT

This paper surveys issues and requirements for future Information Warfare (IW), and introduces our concepts for an area we call: "dynamic information defense" [1]. Although defensive IW would incorporate relatively static information security (INFOSEC) capabilities, an effective IW defense must survive exploitation of pervasive "weak links" in security. This demands countermeasures of a fundamentally more dynamic, cooperative, and distributed nature than are available today. As described in this paper, dynamic information defense transcends INFOSEC with a broader strategy that integrates planning and analysis with a means for situational intelligence to achieve robust in-depth information defense.

1 INTRODUCTION

The information age has brought changes that challenge our ability to ensure the availability, integrity, and security of systems and information infrastructures [2]. New technologies and information needs exceed the state-of-the-art, let alone the state-of-the-practice, in information assurance and information security (INFOSEC). The predominant security models and implementations of the 1980s were oriented toward securing single monolithic systems. In the main, INFOSEC did not anticipate the nature of, and did not meet the security needs for computing in the 1980s. For instance, the development of windowing systems challenged trusted operating systems to maintain the classification levels of documents. Likewise, the rapid rise of networks, desktop computers, and workstations resulted in a decentralization of control over information resources that challenged information security practices and capabilities.

In the 1990s, advances in performance, multimedia, internetworking, and hypertext — combined with the phenomenal appeal of the WWW — have resulted in the seemingly universal desire to interconnect networks in order to disseminate or access information. Recent computing trends have brought further challenges as technology continues to evolve. INFOSEC challenges in the 1990s include meeting requirements that may conflict, such as the need for high-assurance protection, while concurrently simplifying access to information. Similarly, having a means to trust information sources and identities can run counter to the need to assure information privacy.

* We define the term *dynamic information defense* as: An integrated set of automated, flexible countermeasures used to facilitate IW threat detection and to dynamically plan, monitor, and control a range of coordinated responses.

As information infrastructures become increasingly interdependent and complex, we also grow increasingly dependent upon them. These systems have shown vulnerabilities to attack and exploitation [3, 4]. If our information defenses do not evolve to meet continued technological advances, then we will not be able to meet emerging information needs with information infrastructures that can withstand offensive or exploitative threats.

Information Warfare (IW) [5] is motivated by the opportunities that arise from an ever increasing dependence upon vulnerable information systems. IW is the information age battlefront whose scope circumvents physical and electronic defenses which extend throughout the IW realms of Military, Political, Economic, Social, and Physical. Each realm consists of a complex, interdependent infrastructure of systems and processes that are subject to attack and exploitation by a range of adversaries. As shown in Figure 1, each IW realm is based upon the information spectrum—Policy, Physical, Electromagnetic, Infrastructures, and Interoperability. Specific vulnerabilities to a realm occur throughout the information spectrum; therefore, vulnerabilities unique to each piece of the spectrum are subject to attack or exploitation. Regardless of borders or geography, all digital information assets are at least potentially vulnerable to IW threats [6].

IW Realms

Military	Political	Social	Economic	Physical

Information Spectrum

Policy	Physical	Electromagnetic	Infrastructures	Interoperability
- Defense - National - International	- Facilities - People - Procedures - Decision Nodes - Communication	- Power & Telephone - Radio Waves - Microwaves - Infrared - Ultraviolet - X-Rays - Gamma Rays	- Telecommunications - Information Services Information Technology/ Products (Advanced Computing, Information and Networking Technologies) - People (Creation and use of Information Development of Applications and Services,Facilities Construction, and Training)	- Commercial - Government - Joint - Coalition - Intragovernmental

Vulnerabilities to Information Attack, Defense, and Exploitation

Figure 1 — The Information Spectrum and IW Realms

To achieve a specific objective, a given information system may be targeted directly or indirectly. Likewise, in pursuit of tactical goals, an IW attack could exploit the dependency of a targeted system on one or more of its enabling components [7]. IW threat vectors will evolve as processing power, storage capacities, and network bandwidth and connectivity continue to advance.

While a low-technology IW attack only needs to exploit a subset of the vulnerabilities, a medium, or high technology IW attack would likely overwhelm targeted systems and infrastructures. Today, we have only rudimentary, semi-automated, and human-intensive means for countering these threats. While technology which poses IW threats need only be simple and unsophisticated, effective countermeasures are easily orders of magnitude more difficult to implement.

Consequently, there is a clear need for flexible and responsive IW capabilities that form an integrated set of automated countermeasures. These must transcend information defense and should implement the information-age equivalent of the appropriate 'counter' disciplines. Not only will such countermeasures need to facilitate detection, but they must also be able to dynamically affect a range of coordinated defenses. Such countermeasures are themselves prone to exploitation and attack, leading to a cycle that may be similar to counter-counter-escalation in the realm of Electronic Warfare (EW).

The remainder of this paper presents a high-level overview of our concepts and approach for an area we call: "dynamic information defense." Section 2 surveys the basic principles of INFOSEC and presents a brief background on IW. Section 3 identifies the essential issues for a future information war in terms of requirements and technologies. We discuss our concept of "dynamic information defense" and outline the requirements of a strategy for in-depth information defense. These are shown to be significantly broader in scope than static INFOSEC countermeasures. Section 4 outlines our principal research goals.

2 BACKGROUND

While INFOSEC is oriented toward information assurance or protection, IW is by definition more dynamic and demands robust and flexible means for information attack, exploitation, and defense. Today, information defense failures, insufficient mechanisms, and insufficient defense strategies are common in INFOSEC. These defenses are typically static in nature, feature minimal flexibility, offer limited reaction capabilities, and they are typically standalone and not coordinated beyond a narrow range of functionality.

In contrast, on the battlefield, when positional defense fails, a commander has a range of options to include counterattack in order to retake seized ground, or a defense in-depth to not only retake terrain, but to also inflict maximum damage to the enemy by channeling initial attacks into killing zones. Similarly, intelligence officers respond when security is breached by a hostile intelligence services agent, typically by attempting to double the source, thereby turning an otherwise intelligence disaster into an advantage.

To meet the challenges of comparable IW situations requires significant advances in information defense countermeasures. As explained next, although existing INFOSEC countermeasures have a comparatively primitive and narrow range of reaction capabilities, they are necessary within a much broader and augmented defensive IW framework.

2.1 INFOSEC

Briefly, INFOSEC is concerned with protecting information against failure, error, attack, and catastrophe with the goal of preventing denial of service, improper disclosure, modification, or destruction of information. INFOSEC countermeasures are generally oriented toward defending systems from known or somewhat predictable threats. The process of selecting countermeasures is usually driven either by high-level policy or by a cost-benefit tradeoff to assess vulnerabilities and analyze risks.

However, in terms of the threats posed by IW against countermeasures, neither the state-of-the-practice nor the state-of-the-art in INFOSEC are prepared to address the challenges of defense against IW attack. This is because INFOSEC countermeasures, such as trusted operating systems, guards, firewalls, network monitoring, and intrusion detection tend to be:

- Orientated toward known threats or vulnerabilities and tend to address single vulnerabilities, versus being active defenses against new or multiple vulnerabilities that may be exploited in concert;
- Difficult to configure for accurate and reliable operation and typically are not updated in response to changes to the computing environment or threat vectors;
- Functionally limited and inflexible, and rarely include significant information or knowledge about the protected domain. While such capabilities as domain name services, audit-based intrusion detection systems, and network routers maintain more information about their environments, even these are limited in responding to security situations by changing their missions or rule-bases; and
- Lacking all but rudimentary interoperability or information sharing capabilities and rarely leverage situational information from a given domain or exchange threat information with other systems.

These and other limitations, make it impossible to construct an effective IW defense solely on such countermeasures. In an IW campaign, we should expect a maelstrom of threats whose particular form can not be fully anticipated in advance and which would likely change as we reacted to them.

2.2 IW

Development of an effective IW defense can be considered analogous to the development of Command, Control, and Communications Countermeasures (C^3CM) [8]. In the 1970s the Soviets advanced their concept of Radio-Electronic

Combat (REC) [9]; the US response was the development of C^3CM. C^3CM is often advanced as a forerunner of Command and Control Warfare (C^2W) [10,11] — the DoD implementation of IW. It is important to clarify the relative demands of C^3CM (an industrial age, single threat, technology driven concept) vis-à-vis the greater demands of C^2W (a post-cold war, information age vision). First, C^3CM was primarily based on a philosophy that "the best defense is an attack." It was limited in its attack-protect balance. Second, it was oriented on communications as not only a main means of implementation, but as the best one. C^3CM lacked a synergistic and simultaneous approach to information as the key. Lastly, C^3CM addressed the tactical-operational environment during hostilities—but only within the theatre of operations. Little or no consideration was given to pre-hostilities conditioning, post-hostilities requirements, or relevant information intelligence within a global context.

In contrast, C^2W is built upon five pillars and is supplemented by intelligence support, as shown in Figure 2. We recognize the importance of Relevant Information Intelligence (RII) [12], and identify three additional classes of intelligence information as necessary for IW, C^2W, and a dynamic information defense. These classes are:

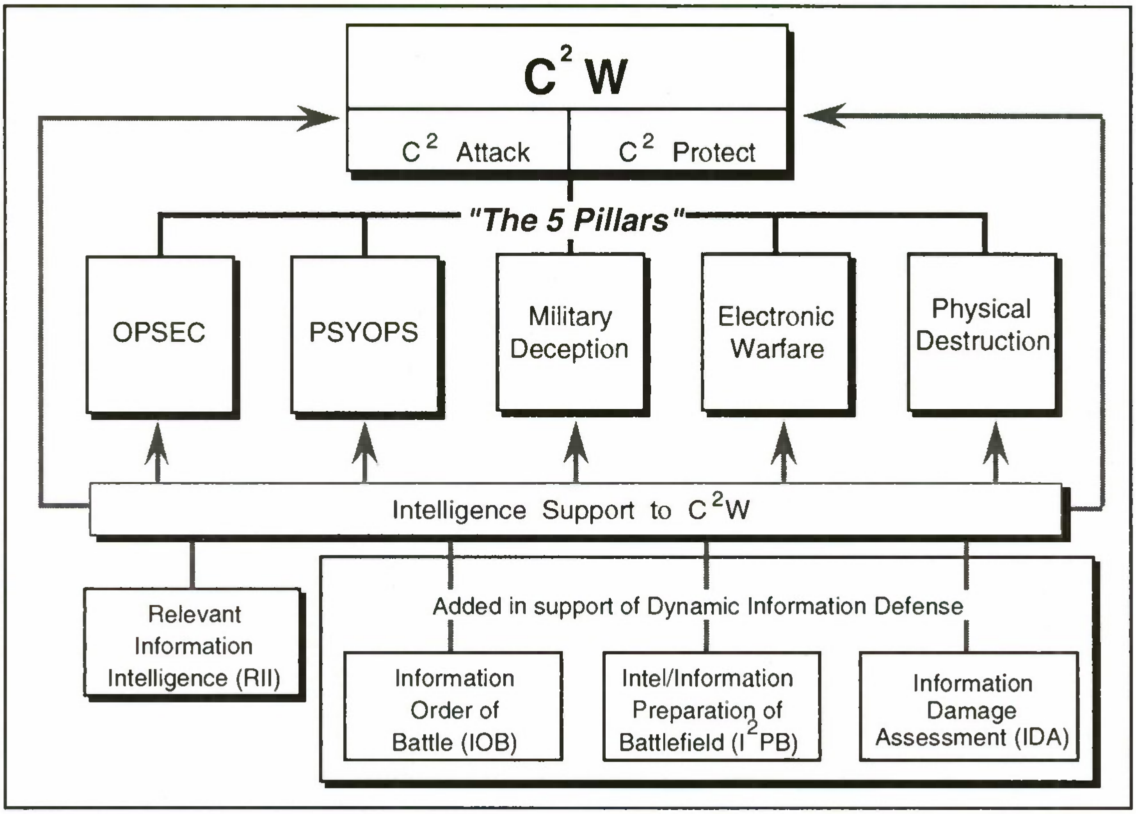


Figure 2 — The Pillars of C^2W

- Information Order of Battle (IOB) — we define IOB as: the command, mission, and information flow structure of any military force as well as all enabling information infrastructures. C^2W, operational security (OPSEC), and targeting in IW often extend beyond the commanders area of influence and thus require a greater degree of coordination at higher levels;
- Intelligence/Information Preparation of Battlefield (I^2PB) — we define I^2PB as: the incorporation of RII and IOB into IPB to enhance the waging of information-based warfare; and
- Information Damage Assessment (IDA) — we define IDA as: the automated identification, assessment, and reporting of information attack or information exploit attempts.

Significant differences exist between C^3CM and C^2W. Just in terms of C^2W objectives, consider the magnitude and relevance of these to the evolution of C^2-attack and C^2-protect. These are to cause or force:

- An adversary to make a substantive decision favorable to exploitation by oneself (e.g., changes in force allocation or plans via disruption or destruction);
- An adversary to make changes in their planned time lines favorable to exploitation by oneself (e.g., delays via disruption, destruction, or manipulation);
- An adversary to make a decision favorable to oneself (e.g., degradation of offensive capabilities in a particular locale via deception or perception management);
- Gridlock in an adversary's decision making capabilities, while our own remain intact (e.g., simultaneity in destruction, disruption, and deception); and
- An adversary into accepting situations or conditions that are contrary to their objectives (e.g., terrorist's imposition of their demands or a nation state's deterrence through information power or some combination of national power employing information).

The information process and the decision/C^2 process [13] are fundamental to achieving the objectives of C^2W. This is done by utilizing the total information spectrum throughout the IW realms, and across the time line that encompasses pre-hostilities, hostilities, and post-hostilities. Just as the information spectrum is not solely dependent upon the electromagnetic spectrum, neither is the military IW spectrum solely dependent upon military assets. In IW, when several threat vectors are used, perhaps in conjunction with Dominant Battlespace Awareness (DBA) targeting, the result can be the overwhelming application of precision force.

From the discussion above, it is evident that the practice of INFOSEC and existing countermeasures are not sufficient to meet the needs of IW or the objectives of C^2W. Survival in an IW theatre demands countermeasures much broader in functionality and more advanced than existing ones.

3 Future IW: Issues and Requirements

Today, a commander's actions can no longer be governed only by what he controls in a theater of operations. He operates in a global infosphere where vulnerabilities to IW attack are spread across all realms. To ensure military success or dominance in IW, we must address this fact. Where information systems are critical—and vulnerable to attack—countermeasures equal to the task need to be in place.

The tempo and scope of an IW attack entails near-real-time (NRT) defense capabilities. Countermeasures need to respond to existing threats, combinations of threats, and emerging threats. Thus, we require countermeasure functionality that can not always be fully defined in advance of attack. In our estimation, IW defense will require countermeasures that are automated, dynamic, flexible, adaptive, and that not only survive but dominate threats. In part, this will require significant advances in computing technology, particularly in such areas as intelligent agents, adaptive systems, and the systems equivalent of OPSEC.

Defensive IW needs to detect, analyze, plan, and control counter attacks. It must be effective despite uncertainties, chaos, and failures that are common in operational situations. A timely, coordinated, and robust response to threats requires a range of command and control functionality that spans centralized, cooperative, and independent operation—throughout the information spectrum and across each IW realm.

3.1 Dynamic Information Defense

The implementation of information assurance throughout the information spectrum requires full counterpart objectives, organization, doctrine, and technology. This can be classified as an in-depth information defense strategy. In contrast to a typical information defense that is vulnerable to, and unlikely to survive compromise of a single weak link, an in-depth information defense strategy includes additional defenses.

We define the term *dynamic information defense* as: an integrated set of automated, flexible countermeasures used to facilitate IW threat detection and to dynamically plan, monitor, and control a range of coordinated responses. Implementing this entails a combination of centralized and distributed IW capabilities to execute the overall information defense mission. Individually, distributed countermeasures would be tasked to mitigate a variety of threats. Thus, we see a need for flexible and intelligent countermeasures, which can satisfy the need for defenses to augment and extend existing INFOSEC countermeasure capabilities.

- Augment existing countermeasures with dynamic and reconfigurable elements for countering threats that are outside the scope of, that would compromise, or circumvent INFOSEC;
- Implement NRT information damage assessment (IDA) or compromise [14];
- Implement a secure means of inter-communication between countermeasures for dissemination of defense plans, situational information, and cooperation;
- Be implemented with both centralized and distributed components—the distributed components would likely include iA or related technology and would be capable of being dynamically tasked according to an OPSEC database or a disseminated information defense plan [15]; and
- Use an OPSEC database to support both the centralized and distributed defense components.

Consistent with C^2W, it may prove necessary to include offensive counter information operations (OFCIOs)—the military equivalent of counterattacking [16] within a defense in-depth area of operations. Within this context, the objectives for OFCIOs would be to:

- Ascertain offensive information operations modus operandi (MO) of adversaries to enhance planning and direction for future information counterattacks;
- Use and redirect an attack to tie-up an adversaries information resources;
- Redirect information attacks to influence and assist friendly operations.

To implement OFCIOs within our dynamic information defense paradigm, we would consider the following factors:

- A reaction course of action (i.e., selection of whether to negate the attack or exploit it through dynamic information defense and specifically OFCIOs);
- A C^2W pillar course of action (i.e., selection of which C^2W pillar will be used for counterattack, for example, disruption of an adversary information system by reversal or deception);
- A time course of action (i.e., whether a counterattack should be immediate or delayed); and
- A damage level course of action (i.e., should a counterattack be gradual or catastrophic).

Clearly, IW is significantly broader in scope than INFOSEC. To a great extent, the range of IW activities are defined by the five C^2W pillars. Our concept of a dynamic information defense is consistent with both IW and the C^2W pillars. This model for a dynamic information defense is a response to the needs of IW defense and the shortcomings of INFOSEC to meet those needs.

4 RESEARCH CONCEPT

Our research focus is on defensive and exploitative IW. The objective is to develop tools to facilitate C^2W efforts under a broader IW campaign. Such capabilities are necessary to counteract an adversary from exploiting, corrupting, and otherwise benefiting from access to our infosphere.

At this time, we have defined the overall project goals and objectives, and developed the functional architecture shown in Figure 4. This architecture is consistent with our information defense in-depth paradigm discussed earlier. We have also begun proof-of-concept prototyping. The principal underlying software technologies include intelligent software agents and Java.

Our prototype is designed to address vulnerabilities in the computing infrastructure and in compromise of critical information that could be exploited. It supports centralized C^2, and features intelligent, automated tools to facilitate planning and analysis for decentralized execution. The prototype is being developed in a distributed, networked environment and features dynamic and flexible IW countermeasures. These are designed to be rapidly reconfigurable to meet and respond to changes in threats. Individual countermeasures may cooperate in pursuit of an overall IW defense as well as in tactical and strategic objectives. For instance, iAs may be deployed among critical nodes, or functional components, that may be associated with or are IW targets. By considering criteria such as risks and vulnerabilities, a component's value as a target to the enemy, and a component's value as an asset to our own warfighters, the decision of when and where to deploy iAs can be made.

Intelligent agents will be used to perform a variety of tasks to defend against IW threats. They will support traditional INFOSEC functions by

Our dynamic information defense paradigm revolves around planning and analysis capabilities. This is driven by the needs of activities such as advance planning, IDA, and countermeasure cooperation. These require planning and analysis and a means to disseminate information associated with these activities. In contrast to the static nature of a traditional INFOSEC vulnerability assessment, IW and dynamic defense activities demand a continuous cycle of information and OPSEC database updating. Information of various classes (such as discussed in Section 2) is required, this includes: RII, IOB, IDA, and I^2PB.

Figure 3 is an overview of our paradigm for dynamic information defense and depicts the perimeters of an information defense in-depth. First, an OPSEC analysis is required to determine known or anticipated vulnerabilities within the information spectrum, the IW realms, and the conflict time-line.

Next, vulnerabilities are addressed with INFOSEC countermeasures. Within a dynamic defense, these countermeasures must become more sophisticated, and should include embedded support for:

- Interoperable encryption as a basic foundation for trusted communications;
- Unforgeable and untamperable identification, for mutual trust, non-repudiation, and OPSEC;
- Untamperable trusted components, including: secure kernels, intrusion detection rule-bases, and security monitoring systems; and

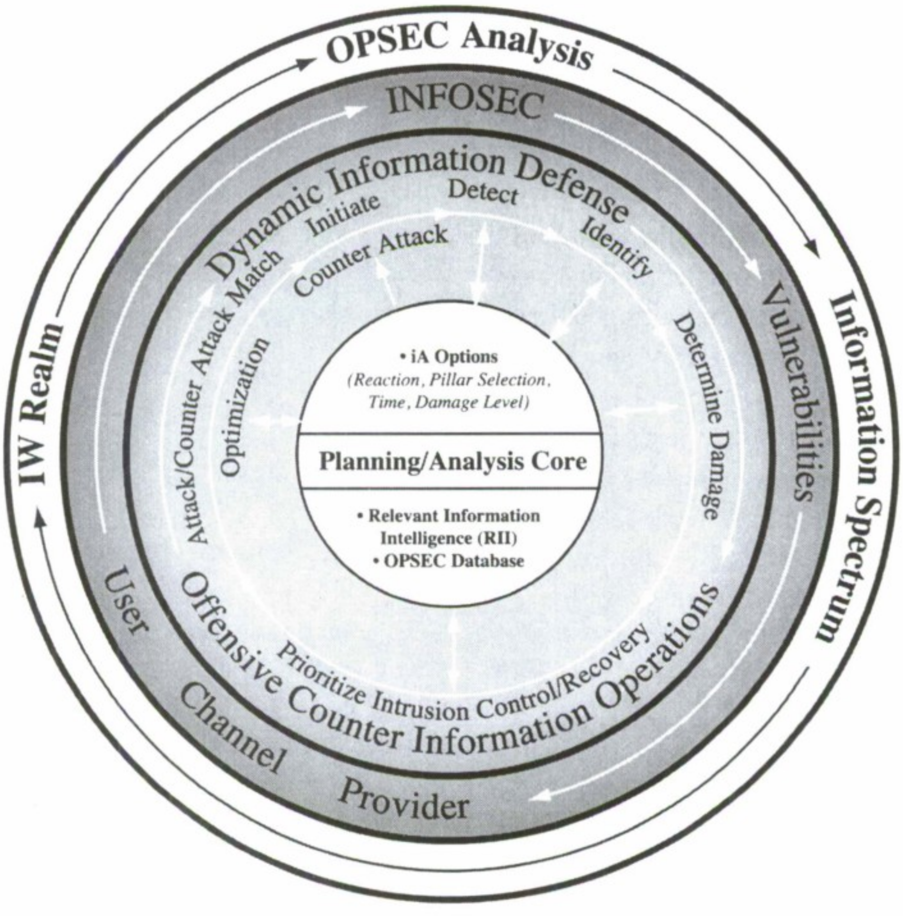


Figure 3 — Information Defense In-Depth Paradigm

- Capabilities for wide-area monitoring of networks, along with a basic means or strategy for the automated generation and communication of situational intelligence.

Dynamic information countermeasures are central to achieving a second and significantly more capable line of defense. While having a partial foundation in INFOSEC, dynamic information defense entails the adaptation of traditional counter disciplines and the use of intelligent components, such as intelligent agents (iAs). In this context, dynamic information defense capabilities would:

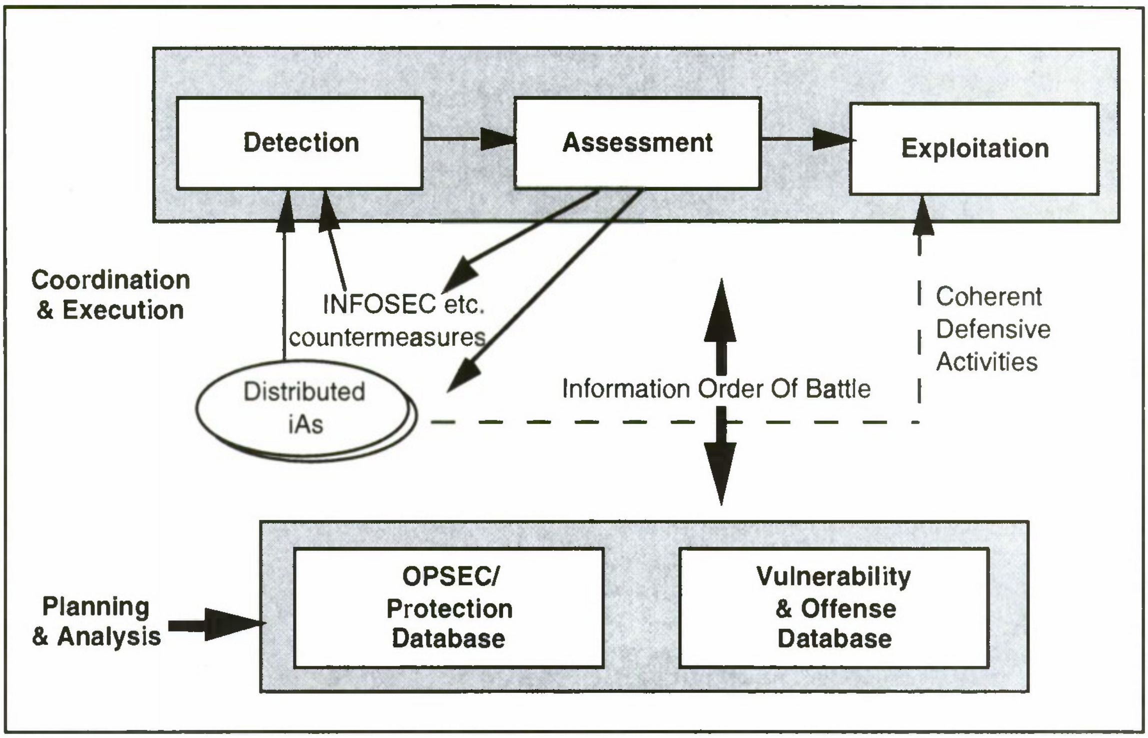


Figure 4 — Functional Architecture for Dynamic Information Defense

performing tasks such as monitoring firewalls and guards, and analyzing network traffic. Such monitoring information then can be leveraged for broader indications and warnings (I&W) and for the dissemination of knowledge about observed IW attack capabilities. This is seen as critical to a coordinated and robust defense. Further, iAs can provide the enhanced capabilities needed by detecting, observing, analyzing, and reporting on previously undefined offensive IW attacks. In response to detected attacks, the iAs may respond:

- Independently in accordance with previously defined scenarios (stored in an OPSEC database);
- In concert with other deployed iAs; and
- In concert with the Central Coordinating Facility (CCF), discussed next.

The final component of the prototype is the CCF, which directly supports IW battle management by:

- Monitoring and displaying IW status;
- Facilitating information damage assessment;
- Providing a dynamic planning and analysis capability to respond to threat situations which could not be fully anticipated or defined in advance of attack;
- Managing the iA knowledge base, which encompasses both the OPSEC database component of previously defined threat response scenarios as well as the database component used to support the dynamic planning and analysis capability;
- Coordinating the execution of responses to detected attacks in concert with deployed iAs; and
- Facilitating centralized reporting of status and lessons learned.

Within this framework, we intend to prototype various concepts and assess their usefulness in counteracting an adversary's attempt to exploit, corrupt, and leverage access to our infosphere. If successful, results of our prototyping activities will make a significant contribution toward empowering the warfighter with the means to effectively manage an IW campaign.

5 Summary

It is essential that our information defenses evolve to meet the continued revolution in technological advances and to provide the US with information infrastructures that are able to withstand offensive or exploitative IW threats. Today, neither the state-of-the-practice nor the state-of-the-art in INFOSEC are prepared to address the challenges of defense against IW attack.

This paper has presented a high-level overview of our concepts and approach for the implementation of a dynamic information defense. Since survival in the IW theatre demands countermeasures that are broader in functionality and more advanced than existing INFOSEC capabilities, our concept integrates planning and analysis into an in-depth information defense. To this end, we have begun development of a prototype for an intelligent, distributed, coordinated, and dynamic information defense capability.

Acknowledgments

The authors owe credit to the ideas and efforts of our colleagues: Justin Landry, Julie Lehnertz, and John Page. In addition, thanks are due to Dr. Jude Franklin and Dr. John White for their continued support.

---

1 Winkler, J.R. O'Shea, C.J., Stokrp, M.C. "Information Warfare & Dynamic Information Defense", June 1996 Command and Control Symposium, Naval Postgraduate School, Monterey CA. June 1996.

2 Alberts, Dr. David S., "The Unintended Consequences of Information Age Technologies", National Defense University, NDU Press Book, April 1996.

3 Swett, Charles, "Strategic Assessment: The Internet", Office of The Assistant Secretary of Defense for Special Operations and Low-Intensity Conflict (Policy Planning), 17 July, 1995

4 Staff of the Security Policy Board "White Paper on Information Infrastructure Assurance", December 1995.

5 IW is defined by the DoD as: "Actions taken to achieve information superiority by affecting adversary information, information-based processes, information systems and computer-based networks while defending one's own information, information-based processes, information systems and computer-based networks" [Draft DODDIR 3600.1, 1996].

6 Libicki, Martin C., "The Next Enemy" National Defense University, Strategic Forum, Number 35, July 1995

7 For instance, the strategic value of an IW attack against one or more of the infrastructures that enable a base-level communications network (such as the power grid, the National Information Infrastructure (NII), the Defense Information Infrastructure (DII), or other non-military communications) could indirectly achieve the equivalent tactical goals as attacking specific computing or communications nodes within the network.

8 Littlebury, F. E.; Praeger, D. K. "INVISIBLE COMBAT: C^3CM - A GUIDE FOR THE TACTICAL COMMANDER." Washington DC: AFCEA International Press; 1986; ISBN: 0-916159-11-6.

9 Hemsley, John. "Soviet Troop Control: The Role of Command and Technology in the Soviet Military System". Oxford, U.K. and N.Y.: Brasseys Limited, 1982.

10 Chairman of the Joint Chiefs of Staff. "COMMAND AND CONTROL WARFARE." ; 1993 Mar 8; Memorandum of Policy No. 30.

11 Chairman of the Joint Chiefs of Staff. "JOINT DOCTRINE FOR COMMAND AND CONTROL WARFARE (C^2W)" (Preliminary Coordination Draft). ; 1995 May; JOINT PUB 3-13.

12 Relevant Information Intelligence (RII) — Current intelligence concerning a potential adversaries information capabilities, systems, dependencies, and the status of information infrastructures within an area of operations.

13 The information process transforms data to information to knowledge. The decision/C^2 process consists of the cycle: observe, orient, decide, act— the OODA loop.

14 The cycle for accomplishing IDA is: detect, identify, determine damage, and prioritize intrusion control and recovery actions.

15 We envision several classes of distributed defensive components that are capable of a range of cooperation and information-sharing in support of information defense. These would serve as the automated

equivalent of a command hierarchy; i.e., centralized control and decentralized execution.

16 The cycle for counterattack implementation is: construction of counterattack options (accomplished prior to attack and continually refined), attack/counterattack match optimization, decision and initiation of counterattack.

Security for Mobile Agents: Issues and Requirements *

William M. Farmer, Joshua D. Guttman, and Vipin Swarup

The MITRE Corporation
202 Burlington Road
Bedford, MA 01730-1420
{farmer,guttman,swarup}@mitre.org

Abstract

Mobile agents are processes which can autonomously migrate to new hosts. Despite its many practical benefits, mobile agent technology results in significant new security threats from malicious agents and hosts. The primary added complication is that, as an agent traverses multiple hosts that are trusted to different degrees, its state can change in ways that adversely impact its functionality. In this paper, we investigate these new threats and develop a set of achievable security requirements for mobile agent systems.

1 Introduction

Currently, distributed systems employ models in which processes are statically attached to hosts and communicate by asynchronous messages or synchronous remote procedure calls. Mobile agent technology extends this model by including mobile processes, i.e., processes which can autonomously migrate to new hosts. This basic idea results in numerous benefits including flexible, dynamic customization of the behavior of clients and servers and robust remote interaction over unreliable networks.

Threats, vulnerabilities, and countermeasures for the currently predominating static distributed systems have been studied extensively; sophisticated distributed system security architectures have been designed and implemented [11, 14]. These architectures use the access control model, which provides a basis for secrecy and integrity security policies. In this model, objects are resources such as files, devices, processes, and the like; principals are entities that make requests to perform operations on objects. A reference monitor is a guard that decides whether or not to grant each request based on the principal making the request, the operation requested, and the access rules for the object.

The process of deducing which principal made a request is called *authentication*. In a distributed system, authentication is complicated by the fact that a request may originate on a distant host and may traverse multiple machines and network channels that are secured in different ways and are not equally trusted [11]. The process of deciding whether or not to grant a request—once its principal has been authenticated—is called *authorization*. The authentication mechanism underlies the authorization mechanism in the sense that authorization can only perform its function based on the information provided by authentication, while conversely authentication requires no information from the authorization mechanism.

Despite its many practical benefits, mobile agent technology results in significant new security threats from malicious agents and hosts. In fact, several previous uses of mobile agents have been malicious, e.g., the Internet worm. The primary added complication is that, as an agent traverses multiple machines that are trusted to different degrees, its state can change in ways that adversely impact its functionality.

In this paper, we will examine a few different ways of using mobile agents, with the aim of identifying many of the threats and security issues which a meaningful mobile agent security infrastructure must handle. We will develop a set of security requirements for mobile agent systems and will distinguish between those that appear impossible, those that are achievable with current technology, and those that might be achievable with future work. We will not, in this short paper, develop a security model which can meet the achievable requirements, though we think it can be done. See [6] for elements of such a model and [4, 9, 13, 15, 16] for related work on mobile agent security.

*This work was supported by the MITRE-Sponsored Research Program.

2 Mobile Agents

A mobile agent is a program that can migrate from one networked computer to another while executing. This contrasts with the client/server model where non-executable messages traverse the network, but the executable code remains permanently on the computer it was installed on. Mobile agents have numerous potential benefits. For instance, if one needs to perform a specialized search of a large free-text database, it may be more efficient to move the program to the database server rather than move large amounts of data to the client program.

In recent years, several programming languages for mobile agents have been designed. These languages make different design choices as to which components of a program's state can migrate from machine to machine. In Java [12], only program code can migrate; no state is carried with the programs. In Obliq [2], first-class function values (closures) can migrate; closures consist of program code together with an environment that binds variables to values or memory locations. In Kali Scheme [3], again, closures can migrate; however, since continuations [10, 8] are first-class values, Kali Scheme permits entire processes to migrate autonomously to new hosts. In Telescript [18], functions are not first-class values; however, Telescript provides special operations that permit processes to migrate autonomously.

The languages also differ in their approach to transporting objects other than agents. When a closure or process migrates, it can either carry along all the objects (mutable data) that it references or leave the objects behind and carry along network references to the objects. Java does not address this issue since it permits only program code to migrate. In Obliq, objects remain on the node on which they were created and mobile closures contain network references to these objects; if object migration is desired, it needs to be programmed explicitly by cloning objects remotely and then deleting the originals. In Kali Scheme, objects are copied upon migration; this results in multiple copies of the same objects; data consistency needs to be programmed explicitly if it is desired. In Telescript, objects can either migrate or stay behind when an agent that owns them migrates. However, if other agents hold references to an object that migrates, those references become invalid. Hence, programming care is required to protect against dangling pointers.

In this paper, we adopt a fairly general model of mobile agents. Agent interpreters run on individual networked computers and communicate among themselves using host-to-host communication services. An agent consists of code together with execution state. The state includes a program counter, registers, environment, recursion stack, and store. Agents execute by being interpreted by agent interpreters.

Agents communicate among themselves by message passing. In addition, agents can invoke a special asynchronous "remote apply" operation that applies a closure to arguments on a specified remote interpreter. Remote procedure calls can be implemented with this primitive operation and message passing. Agent migration and cloning can also be implemented with this primitive operation, using first-class continuation values.

3 Two Examples

In this section, we will describe two examples. We believe they are typical of many—though not of all—of the ways that mobile agents can effectively be used. We will try to draw out the most important security issues that they raise, as a concrete illustration of the problems of secure mobile agents.

Competing Airline Carriers. Consider a mobile agent that visits the Web sites of several airlines searching for a flight plan that meets a customer's requirements. We focus on four hosts: a customer host, a travel agency host, and two servers owned by competing airlines, for instance United Airlines and American Airlines, which we assume for the sake of this example do not share a common reservation system. The mobile agent is programmed by a travel agency. A customer dispatches the agent to the United Airlines server where the agent queries the flight database. With the results stored in its environment, the agent then migrates to the American Airline server where again it queries the flight database. The agent compares flight and fare information, decides on a flight plan, migrates to the appropriate airline host, and reserves the desired flights. Finally, the agent returns to the customer with the results.

The customer can expect that the individual airlines will provide true information on flight schedules and fares in an attempt to win her business, just as we assume nowadays that the reservation information the airlines provide over the telephone is accurate, although it is not always complete.

However, the airline servers are in a competitive relation with each other. The airline servers illustrates a crucial principle: *For many of the most natural and important applications of mobile agents, we cannot expect the participants to trust one another.*

There are a number of attacks they may attempt. For instance, the second airline server may be able to corrupt the flight schedule information of the first airline, as stored in the environment of the agent. It could surreptitiously raise its competitor's fares, or it could advance the agent's program counter into the preferred branch of conditional code. As we will argue in Section 4.1, cryptography does not help here either. Thus, the mobile agent cannot decide its flight plan on an airline host since the host has the ability to manipulate the decision. Instead, the agent would have to migrate to a neutral host such as the customer's host or a travel agency host, make its flight plan decision on that host, and then migrate to the selected airline to complete the transaction. This attack illustrates a principle: *An agent's critical decisions should be made on neutral (trusted) hosts.*

A second kind of attack is also possible: the first airline may hoodwink the second airline, for instance when the second airline has a cheaper fare available. The first airline's server surreptitiously increases the number of reservations to be requested, say from two to 100. The agent will then proceed to reserve 100 seats at the second airline's cheap fare. Later, legitimate customers will have to book their tickets on the first airline, as the second believes that its flight is full. This attack suggests a third principle: *Unchanging components of the state should be sealed cryptographically.*

Distributed Intrusion Detection. Consider an intrusion protection system that protects networked computer systems from electronic attacks by collecting audit data, detecting electronic attacks, and responding to suspected attacks. Mobile agents can be used to dynamically alter the data being collected, distribute the computation across the network, and dynamically respond to suspected attacks. The potential benefits of a mobile agent architecture include greater flexibility and improved performance.

In an ongoing project, we are designing a mobile agent architecture where the network is partitioned into one or more network domains. Each domain has a protected computer running an interpreter that is trusted by all agents within that domain. These interpreters trust each other to varying degrees depending on the relationships between the domains. All other interpreters run on untrusted computers that the intrusion protection system is trying to protect; hence these interpreters cannot be trusted.

The agents of this system will require special privileges to collect audit data and respond to attacks. At the same time, the agents will need to be restricted so that they cannot exceed their authority. An important aspect of this example is that the agents will execute on untrusted hosts in a hostile environment. In order to be effective, the system will require strong security controls to protect both the intrusion detection system and the underlying computer infrastructure.

Numerous attacks, both inadvertent and deliberate, are possible. Intruders can terminate or modify the behavior of interpreters. They can inject their own agents and can modify or trick legitimate agents into performing malicious tasks. They can spy on sensitive data stored within agents, within interpreters, and within communications between agents and interpreters.

Consider a data collection agent that is dispatched by a trusted interpreter, migrates to an untrusted machine, collects process information from that host (e.g., by running "ps" on a UNIX host), then migrates back to the original interpreter to deposit the collected information. If the network addresses of the two interpreters are stored as state variables of the agent, the second interpreter can switch the two addresses, reset the program counter, and return the agent to the first interpreter. The agent will now collect process information from the first interpreter and return it to the second interpreter, thus providing valuable information to an attacker. This attack illustrates that *a migrating agent can become malicious by virtue of its state getting corrupted.*

Ideally, we would like the interpreters to distinguish between agents of the intrusion detection system and agents of attackers. The interpreters should verify the integrity of agents and should execute legitimate agents correctly. The interpreters should provide agents with appropriate resources but prevent harmful behavior. Agents should be able to communicate privately and restrict access to sensitive code or data that they carry. Agents should execute correctly and completely; that is, agents should migrate correctly to desired hosts, execute correctly on those hosts, and should be recovered in the event of system failure.

4 Security Goals

Security is a fundamental concern for a mobile agent system. Harrison et al. [7] identify security as a "severe concern" and regard it as the primary obstacle to adopting mobile agent systems.

The operation of a mobile agent system will normally be subject to various agreements, whether declared or tacit. These agreements may be violated, accidently or intentionally, by the parties they are intended to serve. A mobile agent system can also be threatened by parties outside of the agreements: they may create rogue agents; they may hijack existing agents; or they may commandeer interpreters.

There are a variety of desirable security goals for a mobile agent system. Most of these concern the interaction between agents and interpreters. The user on behalf of whom an agent operates wants it to be protected—to the extent possible—from malicious or inept interpreters and from the intermediate hosts which are involved in its transmission. Conversely, an interpreter, and the site at which it operates, needs to be protected from malicious or harmful behavior by an agent.

Not all attractive goals can be achieved, however, except in special circumstances. In the case of mobile agents, one of the primary motivations is that they allow a broad range of users access to a broad range of services offered by different—frequently competing—organizations. Thus, in many of the most natural applications, many of the parties do not trust each other. In our opinion, some previous work (for instance [16]) is vitiated by this fact: It assumes a degree of trust among the participants which will not exist in many applications of primary interest.

Nevertheless, the special cases may be of special interest to some organizations. A large organization like the United States Department of Defense might set up a mobile agent system for inter-service use; administrative and technical constraints might ensure that the different parties can trust each other in ways that commercial organizations do not. In this paper, however, we will focus on the more generic case, in which there will be mistrust and attempts to cheat.

To emphasize the consequences of this choice, we will first discuss putative security goals that we believe cannot be achieved in realistic cases. We will then turn to the security services that can already be supported by well-known techniques for security in distributed systems. Finally, we will identify some security goals that we believe can be achieved, but not without novel additions to current distributed security mechanisms.

4.1 What is Impossible

Several apparently desirable security goals appear unachievable in the generic case we are focusing on.

Is an interpreter untampered? There appears to be no reliable way to authenticate an interpreter. For instance, suppose that one wants to determine whether the interpreter running on a particular host has been tampered with, in the sense that its text segment does not match a given executable image identically. In case the host is not running an operating system that one trusts, there appears to be no way to ensure this.[1]

In our context we can assume that many of the hosts will be purchased and maintained by adversaries, or at least competitors. Then, first, the host is unlikely to allow one to log in and inspect the memory of the running executable to do the comparison by hand, so to speak. Second, a utility program running on that host to perform such comparisons on our behalf could itself have been tampered with, leading to a regress. Third, it is infeasible in general to determine, by sending test scripts, whether an interpreter has been tampered with; the tampering has probably been designed to be unobtrusive, and to make a difference only in odd but important circumstances. Testing software is hard enough in a non-adversarial context; bugs may survive lengthy testing even if they were not designed to be hard to find.

Will an interpreter run an agent correctly? Programs are merely a special kind of data, and agents are merely itinerant programs with some additional types of data attached. Because the agent is essentially passive, there is no way to ensure that the interpreter will execute the program in accordance with the intended semantics of the program. Moreover, there is normally no way to check whether an agent has been executed faithfully: If we knew what result it would compute, we would not have needed to send the agent.

It may sometimes be possible to determine heuristically that an interpreter is cheating, by sending agents whose results we believe we can predict ahead of time. However, as we mentioned, clever cheaters are apt to escape detection for a long time.

Will a host run an agent to completion? A host may decide, for reasons of its own, to stop execution of an agent.

Will a host transmit an agent as requested? A host may decide, for reasons of its own, not to transmit an agent that requests to move, or alternatively, to transmit it to the wrong destination. However, with suitable public-key cryptographic support, it is possible to ensure that a user is not tricked into thinking that a particular host was contacted if it was not.

[1] On the other hand, if one does have some assurance about the host hardware and operating system, then one can ensure that a valid version of a program will be running [11, Section 6].

Can an agent's code and data be kept private? Since an agent's code must be executed by a potentially large group of interpreters, it must be readable by all of them. Hence, there is little point in attempting to protect it by encryption. A similar point holds for data carried by the agent that will be needed later in its travels; if an agent will need to consult data in its state at an interpreter that its sender does not trust, then that data cannot be encrypted.

By contrast, data an agent has collected may be encrypted with its sender's public key if the data will not be examined again until the agent returns home. If a host may be trusted to provide true data on a particular subject, then this method may be used to ensure no host visited later will be able to change the results meaningfully.

If a pair of interpreters trust each other at least to a limited extent, then they can choose a session key for communications between themselves [11]. In this case they can offer link security to agents: agents being transferred between those interpreters will be transmitted in encrypted form.

Can an agent carry a key? For similar reasons, an agent cannot carry its own key (or other secrets, such as credit card numbers) in a form that can be used on untrusted interpreters. Someone will peek.[2]

A secret such as a key can be carried in *encrypted* form, but an interpreter must be entrusted with a "master key" if the agent is to be able to use the decrypted secret.

However, it appears undesirable to give an agent an encrypted key even for use on trusted interpreters. It is useless until we authenticate an interpreter and distribute the master key on a secure channel, for instance using the interpreter-to-interpreter encryption mentioned above. What point does it serve then to have the agent carry an encrypted key? It seems simpler and more robust to use the interpreter-to-interpreter encryption itself, so long as the agent has a name that the sender can tag the message with. If the interpreter can be trusted with a master key, then it can surely be trusted to give the name correctly over the secure channel.

[2] For this reason we expect, in the example of the airline reservation system, that the agent will make a *reservation* rather than an actual *purchase*. The purchase itself can be handled more safely by having the sender separately engage in an electronic purchase protocol. Such protocols require the purchaser to be on-line—and to demonstrate possession of a private key—as the transaction occurs, unlike mobile agents, which can be active while their sender is off-line.

For an overview of electronic purchase protocols, see `http://www.ini.cmu.edu/NETBILL/commerce.html`.

Can agent-to-agent communication be kept private? Similar considerations apply to agent-to-agent communication. It seems pointless to give agents keys so that they can have authenticated or secret communication with other agents. That mechanism could work only while the agents are executing on trusted interpreters. And in that case, we can use the simpler and more robust interpreter-to-interpreter secure communication. The sending agent passes data to its interpreter, which sends the data through an encrypted channel to the interpreter executing the receiving agent. The interpreters are then trusted to identify the sender and recipient correctly, and to protect the message by proper encryption.

Can an agent be distinguished from a clone? Many mobile agent languages allow agents to clone themselves. However, the system cannot reliably distinguish the original agent from its clone. This is because agents do not carry keys. Thus, if the code and data of the clone are to be authenticated, they must have the same cryptographic checksum as the original agent, as the private keys of the sender and author are not available to construct new ones. Thus, the code and signed data of the clone must be identical to the original. Thus, to distinguish them at all, we must examine the unsigned portion of their state, and there is no guarantee that these components have not been tampered.

4.2 What is Easy

Some fundamental security goals can be achieved by familiar techniques for distributed security.

Can the author and the sender of an agent be authenticated? The identity of the author of the program contained in an agent can be determined if the author signs the code. Similarly, the sender of an agent may make his identity known by signing the program together with such other components of an agent as will remain fixed through its travels. This assumes a certificate validation system; the certificates can migrate with the agents.

Can we check the integrity of an agent's code? Modification of an agent's code can be detected by checking the author's signature.

Can interpreters ensure agent privacy during transmission? Unauthorized parties can be prevented from reading sensitive information held by an agent while it is in transit between two interpreters if the interpreters are willing to encrypt it for transmission.

Authorization: Can interpreters protect themselves against agents? An interpreter (or a remote resource manager) can decide if an agent should have access to a resource by considering the agent's author, program, user, and state. Some of these items may be known to be worthy of a certain degree of trust.

4.3 What is Possible but not Easy

Some security goals cannot be achieved via existing approaches to security for distributed systems. Nevertheless, it appears that they can be achieved by developing special techniques for security in mobile agents. We consider these areas to be the natural context for research in mobile agent security.

We will group the issues into two classes: those which allow an interpreter to evaluate the safety of code that it is to execute, and those which allow an interpreter to evaluate the safety of an agent's state.

Can we use a language in which all programs are safe? One possibility is to develop "safe" languages, in which agents or mobile code have restricted access to operations that affect the environment; Safe-Tcl is an example [1]. In this approach, an incoming, untrusted piece of code is provided with a subset of the language primitive operations; presumably, anything that can be done with these is "safe enough." This approach is reasonable in some contexts, although its flexibility is limited.

Java [13] and Telescript [15] both use aspects of their object oriented programming languages to allow libraries to offer a secure interface to incoming code. The languages are complex, however, and widespread review is only beginning [5]. Undoubtedly piecemeal revisions will be needed, and more importantly, a comprehensive understanding of the semantics of the languages is called for. A good semantics should allow a programmer to draw confident conclusions about what possibilities are allowed by the interface he offers.

Java also offers a *byte-code verifier* [13]. This is intended to check programs at load time. Java code is compiled into an intermediate form called byte-code before it is transmitted. The byte-code verifier is intended to assure an interpreter that a newly arrived piece of byte-code—which may have been compiled by a faulty or malicious compiler—satisfies the same type-correctness properties that a correct compiler would enforce. As far as we know, there has been little independent analysis of its design or implementation.

Can a sender restrict his agents flexibly? In some applications, a sender wants his agent to run with restricted authority in most cases, but with greater authority in certain situations. For instance, in the intrusion detection tool mentioned above, a data-collection agent executing **ps** on an untrusted UNIX system needs only ordinary privilege. However, when it returns to its home interpreter, the agent must request privilege so that it can install the newly gathered information into a protected database. Thus, there must be a mechanism to allow an agent to request different levels of privilege depending on its state (including its program counter).

Can an interpreter ensure that an agent is in a safe state? Because a migrating agent can become malicious if its state is corrupted, as in the case of the intrusion detection **ps** agent, an interpreter may want to execute a procedure to test whether an agent is in a harmful state. However, the test must be application-specific, which suggests that reputable manufacturers of mobile agents may want to provide each one with an appropriate state appraisal function to be used each time an interpreter starts an agent. The code to check the agent's state may be shipped under the same cryptographic signature that protects the rest of the agent's code, so that a malicious intermediary cannot surreptitiously modify the state appraisal function.

Can a sender control which interpreters have authority to execute an agent? If executing an agent involves contacting other hosts, then an interpreter may have to authenticate that it is a legitimate representative of the agent. The sender of an agent may want to control which interpreters will be able to succeed in authenticating themselves in this role.

5 Conclusion

Many of the most important applications of mobile agents will occur in fairly uncontrolled, heterogeneous environments. As a consequence, we cannot expect that the participants will trust each other. Moreover, interpreters may disclose the secrets of visiting agents, and may attempt to manipulate their state.

Existing techniques, intended for distributed systems in general, certainly allow substantial protection within the broad outlines of these constraints. However, substantial investment in mobile agent systems may await further work on new security techniques specifically oriented toward mobile agents. These new techniques, discussed in Section 4.3, focus on two areas. One is programming language support to improve the safety of mobile code. The other is support for tracking the state carried by mobile agents. With advances in these areas, we believe that mobile agents will be an important ingredient in producing secure, flexible distributed systems.

References

[1] N. S. Borenstein. Email with a mind of its own. In *ULPAA '94*, 1994. `ftp://ftp.fv.com/pub/code/other/safe-tcl.tar.gz`.

[2] L. Cardelli. A language with distributed scope. In *Proceedings of the 22nd ACM Symposium on Principles of Programming Languages*, pages 286–298, 1995. `http://www.research.digital.com/SRC/Obliq/Obliq.html`.

[3] H. Cejtin, S. Jagannathan, and R. Kelsey. Higher-order distributed objects. *ACM Transactions on Programming Languages and Systems*, 17(5):704–739, September 1995. `http://www.neci.nj.nec.com:80/PLS/Kali.html`.

[4] D. Chess, B. Grosof, C. Harrison, D. Levine, C. Parris, and G. Tsudik. Itinerant agents for mobile computing. *IEEE Personal Communications Magazine*, 2(5):34–49, October 1995. `http://www.research.ibm.com/massive`.

[5] Drew Dean and Dan S. Wallach. Security flaws in the HotJava browser. Technical Report 95-501, Department of Computer Science, 1995. URL ftp://ftp.cs.princeton.edu/reports/1995/501.ps.Z.

[6] W. Farmer, J. Guttman, and V. Swarup. Security for mobile agents: Authentication and state appraisal. In *Proceedings of the European Symposium on Research in Computer Security (ESORICS)*, September 1996.

[7] C. G. Harrison, D. M. Chess, and A. Kershenbaum. Mobile agents: Are they a good idea? Technical report, IBM Research Report, IBM Research Division, T.J. Watson Research Center, Yorktown Heights, NY, March 1995. `http://www.research.ibm.com/massive`.

[8] C. Haynes and D. Friedman. Embedding continuations in procedural objects. *ACM Transactions on Programming Languages and Systems*, 9:582–598, 1987.

[9] IBM Corporation. Things that go bump in the net. Web page at `http://www.research.ibm.com/massive`, 1995.

[10] IEEE Std 1178-1990. *IEEE Standard for the Scheme Programming Language*. Institute of Electrical and Electronic Engineers, Inc., New York, NY, 1991.

[11] B. Lampson, M. Abadi, M. Burrows, and E. Wobber. Authentication in distributed systems: Theory and practice. *ACM Transactions on Computer Systems*, 10:265–310, November 1992. `http://DEC/SRC/research-reports/abstracts/src-rr-083.html`.

[12] Sun Microsystems. Java: Programming for the internet. Web page available at `http://java.sun.com/`.

[13] Sun Microsystems. HotJava: The security story. Web page available at `http://java.sun.com/doc/overviews.html`, 1995.

[14] J. G. Steiner, C. Neuman, and J. I. Schiller. Kerberos: An authentication service for open network systems. In *Proceedings of the Usenix Winter Conference*, pages 191–202, 1988.

[15] J. Tardo and L. Valente. Mobile agent security and Telescript. In *IEEE CompCon*, 1996. `http://www.cs.umbc.edu/agents/ security.html`.

[16] C. Thirunavukkarasu, T. Finin, and J. Mayfield. Secret agents — a security architecture for KQML. In *CIKM Workshop on Intelligent Information Agents*, Baltimore, December 1995.

[17] T. D. Tock. An extensible framework for authentication and delegation. Master's thesis, University of Illinois at Urbana-Champaign, Urbana, IL, 1994. `ftp://choices.cs.uiuc.edu/Papers/Theses/MS.Authentication.Delegation.ps.Z`.

[18] J. E. White. Telescript technology: Mobile agents. In *General Magic White Paper*, 1996. Will appear as a chapter of the book Software Agents, Jeffrey Bradshaw (ed.), AAAI Press/The MIT Press, Menlo Park, CA.

Extended Capability: A Simple Way to Enforce Complex Security Policies in Distributed Systems

I-Lung Kao
Networking Software Division
IBM Corporation
Austin, Texas
ikao@austin.ibm.com

Randy Chow
Department of CISE
University of Florida
Gainesville, Florida
chow@cis.ufl.edu

Abstract

Capability has been widely used as a fundamental mechanism for access control in distributed systems. When an object server receives a capability from a subject for accessing an object, it verifies the validity of the capability and checks whether the access request is allowed with the access rights placed on the capability. Capabilities have been recognized to be more suitable than centralized access control lists for object protection in a distributed system. However, most existing capability-based systems can only enforce *static* access control policies, which mean all the access privileges a subject possesses for an object are fully represented by a capability and will not change due to object access. However, the security policies required by many complex applications are *dynamic* by its virtue. That is, each access authorization depends upon the subject's access history and/or the object's history of being accessed. This paper proposes an *extended capability* system for enforcing this type of dynamic security policies. The key research issues are how to capture the dynamic access information in both capabilities and object servers while avoiding the disadvantages of using access control lists. Some examples are used to demonstrate the flexibility of the proposed system for enforcing complex policies. The problems regarding capability management including propagation, revocation, and distribution of capabilities are also discussed.

1 Introduction

First proposed by Dennis and Van Horn [4], capability has been used as a fundamental mechanism for object naming and access privilege representation in many protection systems [12, 13, 21]. In general, a capability is just like a ticket, on which the name (logical address) of an object and the access privileges possessed by the holder of the capability are recorded. When a cleint attempts to perform an operation on an object, it presents the corresponding capability for the object to the object server. If the operation requested is allowed by the access privileges shown on the capability, the object server will perform the access on behalf of the client; otherwise, it will deny the access.

Capability in the user space

In traditional centralized operating systems, capabilities are created and managed only by the kernel and stored in the system space. So protection of capabilities from tampering is done by any mechnisms protecing the system kernel. However, this layer of protection for capabilities does not exist any more after the emergence of microkernel-based distributed operating systems. In comtemporary distributed operating systems [7], a capability is created by some trusted object server (it needs to be trusted because it runs in the user space), and then passed to the client and manipulated in the user space of the client. In order to prevent a capability to be forged at will, a cryptographic technique for the integrity of the capability must be employed [22]. That is, a *check* field, which is usually the result of a cryptographic function (computed by the object server), is added to the capability, and only the object server can validate this field. This non-system-controlled capability-based framework has become an attractive approach to the design of modern distributed operating systems [7]. Capabilities are no longer under the tight control of the operating system kernel, and instead are manipulated directly by user processes and incorporated into various mechanisms for object access (e.g., a parameter in a remote procedure call can be reserved for capabaility).

Identity-based capability

A disadvantage of traditional capability, shown by Boebert [3], is that it cannot be used to enforce the *-property of the multilevel security policy [15], mainly due to the property that *"the right to exercise access carries with it the right to grant the access"*. Thus it is very possible that a capability be propagated across domains of subjects at different levels without being detected, and subsequently causes unauthorized accesses [10]. To overcome this problem, Gong [5, 6] suggested to incorporate identities of subjects into traditional capabilities, and to emphasize on checking of capability propagation.

Capability is better in distributed systems

In a distributed system, capability is actually a more suitable mechanism for object protection than *access control list* (ACL) which many currnet operating systems still use, because of several reasons. The first one is performance. In a capability-based system, an object server only needs to validate a capability upon an access request. An ACL-based system, on the other hand, requires much higher overhead due to the searching and checking of an entire access control list, which could be very long in a large distributed environment. Even if the average access control list is short or some variation method (e.g., the protection bits in UNIX) is used, a capability-based system still has performance advantages since the most time-consuming I/O task is performed by each client (to retrieve a capabiltiy) rather than by a possibly heavily loaded object server (to load an access control list) in an ACL-based system. Secondly, a capability-based system is more scalable in the sense that each access authorization is independent of the size of the system. Furthermore, for the purpose of separating policies and mechanisms, modern operating systems usually centralize all access control policies in an *authorization server* and require that all object servers be restricted to contain only access control rules and mechanisms to enforce these policies. Distributed and local checking of capabilities by object servers is more adaptive to such an environment, since otherwise, either each object server needs to inquire the authorization server for each access request or the whole authorization information needs to be duplicated on each object server. With these benefits, it is not surprising that most modern operating systems use capabilities for access control (to name a few: Accent[17], Mach[18], and Amoeba[22]). Apparently, the management of capabilities in an efficient and secure way is an important topic of contemporary distributed systems.

Why a new capability system

Most existing capability-based systems can only enforce *static* access control policies, which mean all the access privileges a subject possesses for an object are represented by the capability for that object and will not change due to an access operation. However, the security policies required by many complex applications are *dynamic* by its virtue [8]. That is, each access authorization may depend upon the subject's access history and/or the object's history of being accessed. This type of dynamic access control policies are difficult to enforce using a conventional ticket-type capability scheme, without resorting to additional access control mechanisms. The concept of access control lists could certainly be modified for enforcing these complex security policies due to its centralized feature. Yet, using centralized access control lists excessively in distributed systems apparently loses the advantages of using capabilities we mentioned above. Accordingly, an extension of the capability-based system to handle complex and diversified security requirements is justified.

This paper proposes an *extended capability* system, which provides additional functions to satisfy many complex security requirements with minimum overhead. The innovative idea is to place complicated and tedious access control information on the extended capabilities distributed to subjects, and to maintain simple and regulated capability processing rules and very little information about objects in object servers. In the following, we first introduce some basics of an extended capability in section 2. Then, three frequently desired policies are used to demonstrate how complex access meditation can be achieved with this capability system in section 3. Finally, some capability management issues are elaborated in section 4.

2 Extended Capability

This section describes the basic assumptions in a general distributed system environment, and the definition and generation of an extended capability.

2.1 System Environment

An object system model is assumed. Each object in the system is encapsulated and managed by an *object server*. An access request to an object can be serviced only by its object server and is authorized by the extended capability presented a subject. The object server is responsible for all the activities regarding capability including generation, distribution, verification, and revocation of capabilities. Each object server is assumed to be a part of the trusted computing base (TCB), which guarantees that the server cannot be bypassed for any access attempt. For brevity, an extended capability described below will be named an *e-cap*.

2.2 Format of an *e-cap*

The format of an *e-cap* and the definitions of all the fields contained are shown in Figure 1. As an identity-based capability [6], an *e-cap* can only be used by the *subject* specified in the capability. Thus, if a suitable authentication mechanism is employed for object access in the system, a malicious subject cannot gain access to the object with a stolen *e-cap*. The *subject* field is further divided into two subfields, the *id* and *type* of the subject. The *rights* field of an *e-cap* determines the access privileges the *subject* possesses to the object and its interpretation depends upon the type of the object. An *e-cap* also has a *lifetime* field which tells when the capability will expire, based on the local clock of the object server. An *ACI* field is used by the object server to store important access control information. It is the primary control for enforcing complex security policies. This field also has different meanings for different types of objects and different policies, and is recognizable only to the object server. The last field of an *e-cap*, *check*, as uaual, is used to protect the capability from forgery or tampering.

2.3 Generation of an *e-cap*

Each object in an extended capability system is associated with a unique secret number, called *seed*, known only to the object server managing the object. The main purpose of the *seed* number is to prevent capability forgery and to facilitate full capability revocation. The secrecy of the *seed* number is crucial to a capability system, so must be fully protected by the object server.

An *e-cap* is created upon the request of a subject to the object server, which then consults the *access control server*, a trusted component where all security policies are maintained. The access control server determines the values of all fields except the *check* in the *e-cap*, according to a specific security policy. These values are returned to the object server which then computes the *check* field to complete the construction of the *e-cap*. The *check* field is computed by using a publicly known one-way function f as follows:

$$check = f(subject, seed, rights, lifetime, ACI)$$

It is actually a signature of the object server on the *e-cap*, and this field will be examined each time the *e-cap* is presented to the server later. The principle of separation of policies and mechanisms is achieved by having the access control server provide the programming interface for specifying security policies as well as the functions to translate these policies to *e-cap* fields. Thus, the object server is only responsible for policy enforcement.

Before we discuss other problems regarding management of *e-cap*'s, we now elaborate how the *e-cap* system can be used to mediate object access beyond the traditional ticket-like scheme.

3 Access Mediation with an *e-cap* System

When an *e-cap* is presented to an object server along with a request to access an object, the server first needs to check whether the *e-cap* has ever been tampered by recomputing the *check* field. Only the subject presenting an *e-cap* with a correct *check* field will "possibly" gain access rights shown on the capability. Then the object server utilizes the information stored in the *ACI* field to determine whether the access attempt should be allowed or denied. Several different ways of utilizing this field are demonstrated below.

3.1 Strongly Typed Systems

In a strongly typed system, every subject and object has a type associated with it and its type cannot be changed discretionally. The type of a subject usually represents the role or class of the subject, and each type often implies a different set of access privileges. The type of an object indicates the category of the information stored in the object.

One *e-cap* for all the subjects of one type

The access patterns of many applications may have the property that all the subjects of one type share the same set of access rights to an object. For example, all the faculty in a Computer Science department have "read" and "write" rights for the department's "Technical_Report_List" file, for which all the students only have a "read" right. Since the

subject	rights	lifetime	ACI	check

subject - The "id" and "type" of the subject who owns the capability

rights - Access rights with bit pattern depending on the type of the object

lifetime - The time when the capability expires

ACI - Access control information sepcified by the access control server

check - Bit field for protecting the capability from forgery

Figure 1: The format of an *e-cap*

subject field of an *e-cap* contains a *type* subfield, we can use it to generalize an *e-cap* such that all the subjects in one type can use the same *e-cap* to access the object. When such a capability is created, the *id* part in *subject* is set all 0's and the *type* of those subjects is specified. The *ACI* field is configured to indicate that this *e-cap* is a *typed* one, thus an access request will be allowed as long as the accessing subject belongs to the *type* and the access operation requires only some or all of the *rights*.

A typed *e-cap* has storage advantage since it can be shared by all the subjects of the same type, thus the need of memory space for storing one capability for each subject can be diminished. Alternatively, it can be freely copied from one subject to other subjects of the same type, so the workload of generating capabilities for all the subjects, especially when the number of subjects in that type is large, by the access control server, can be significantly reduced. To support such a typed *e-cap*, the authentication service needs to ensure that an object server is able to gets the correct type of an accessing subject, which is easily achievable by just including the subject's type in the authentication message.

One *e-cap* for all the objects of one type

Similarly, in a strongly typed system there exist applications requiring that a subject has the same access rights to all the objects of the same type. For example, a professor may have "execute" rights for all the executable files of the "Student_Project" type, and a student has "read" rights for all the files of type "Project_Assignment". To make things easier in such cases, we expect that a subject can use only one capability to access all the objects in the same type. In order to achieve this, the *seed* number associated with an object is augmented to contain two seed numbers, one for the object itself (called *id_seed*) and the other for the type of the object (called *type_seed*). The *id_seed* is still unique to each object, yet all the objects of one type share the same *type_seed*. When an *e-cap* is created, it is the *type_seed* that is used in computing the *check* field with the one-way function, and the *ACI* field is configured to indicate such a capability preparation. Later, when this *e-cap* is presented to the object server, it can be used by the *subject* to access any object in the same type with the *rights* specified in the capability. This technique not only reduces the computation overhead of generating one capability for each object in the same type by an object server, but also saves memory space required to store capabilities by a subject.

3.2 Implementation of n-time Tickets

Some applications require that a group of subjects can only access a particular object for a certain number of times. That is, each subject in the group has a pre-determined number of times to access an object and will not be able to access the object after all of its allowed accesses are done. A special case of this policy is a one-time ticket, by which each subject in a group can only access an object only once. It is apparent that many activities in the real world need this feature. Therefore, we first show how such an access control requirement can be enforced by an *e-cap* system.

Implementing a one-time ticket

Implementing a one-time ticket for each subject in a group can be achieved by using a salient feature of prime numbers, which has been employed to reduce the overhead of manipulating access control lists [14]. Assume the group consists of k subjects, represented as $S_1, S_2, \cdots, S_k$, and each of them will be given an *e-cap* that can be used only once to access an object O. This can be fulfilled by storing a unique prime p_i in the *ACI* field of the capability given to S_i, and by storing a number $prod(O)$, which is the product of all primes (i.e., $prod(O) = p_1 \cdot p_2 \cdots p_k$), with O. When an S_i attempts to access O with its *e-cap*, the number $prod(O)$ will be divided by p_i. If it is divisible, the access request of S_i will be granted and the resulting quotient will become the new $prod(O)$. If not, the access request of S_i request will be denied and nothing changes. Due to the property of primes, $prod(O)$ can be divisible by each p_i only once, which exactly renders a one-time access of O to each S_i.

After all S_i's have accessed O, $prod(O)$ becomes one. If desired, $prod(O)$ can be reset to the initial product number, advised by some processing rule in the object server, thus making each one-time *e-cap* usable once more. Another advantage is its flexibility, in that a new subject S_{k+1} can be added to the group at any time, as long as it is given an *e-cap* with the *ACI* containing a unique prime p_{k+1} and the current $prod(O)$ is multiplied by p_{k+1}. Similarly, a subject S_i can also be removed from the group any time by just dividing $prod(O)$ by p_i.

Extension to n-time tickets

The technique of implementing one-time tickets for a group of subjects to access an object can be extended to a more general case, n-time tickets. That is, each subject S_i is allowed to access O for n_i times, $1 \leq i \leq k$, where each n_i is not necessarily the same. For this case, each S_i is still given an *e-cap* with a unique prime p_i in its *ACI*, but the $prod(O)$ with object O is computed initially as

$$prod(O) = p_1^{n_1} \cdot p_2^{n_2} \cdots p_k^{n_k}$$

The same division operation is performed when subject S_i presents its *e-cap* to the object server along with its access request. Because of the property of primes, $prod(O)$ can be divisible by p_i for only n_i times, which means the *e-cap* of S_i can be valid for only n_i times of accesses. For example, a group of three subjects S_1, S_2, and S_3 can access object O for three, one, and two times, respectively. Assume $p_1 = 2$, $p_2 = 3$, $p_3 = 5$, then initially $prod(O) = p_1^{n_1} \cdot p_2^{n_2} \cdot p_3^{n_3} = 2^3 \cdot 3^1 \cdot 5^2 = 600$. After S_3 accesses O once, $prod(O)$ becomes $\frac{600}{5} = 120$. After S_1 accesses O twice later, $prod(O)$ becomes $\frac{120}{2^2} = 30$, which leaves each S_i only one time of access. In addition to possessing the same advantages as those from one-time tickets, this generalized scheme is even more powerful and flexible, in that it allows the object server, according to the application requirement, to dynamically increase or decrease the number of times a subject can access an object at any time, by appropriately adjusting the value of $prod(O)$.

To implement the n-time tickets for object O, the object server needs to be equipped with some capability processing rules and mechanisms (e.g., generating prime numbers), but only a $prod(O)$ and an index indicating the largest prime used up to now need to be kept for the object.

3.3 Enforcing an Access Sequence

Many business applications have the security requirement that a set of related subjects need to access an object in a specific order with probably different access rights. The *e-cap* system can also support such a requirement with additional functions added to the object server. The idea is to give each subject a different *e-cap* such that a capability can be used to access the object only if each subject strictly follows the pre-determined access sequence. Instead of elaborating how this scheme works generally, an example is used to demonstrate the idea.

Generating capabilities

Let's assume that an object O needs to be accessed by three subjects with different access rights, in a sequence as $S_1 \rightarrow S_2 \rightarrow S_3$. When this access control policy is specified through the access control service, an access sequence number (*ASN*) is assigned to this particular policy. When the object server of O generates capabilities for this policy, this *ASN* will be stored in the *ACI* field of the *e-cap* given to each subject. In addition to the one-way function used to compute the *check* field in an *e-cap*, another one-way function is used by the object server to "simulate" the change of the *seed* number of O for a specific access sequence. These two one-way functions are distinct since their input parameters are different.

1. $f_{check}()$: is the original one-way function to compute the *check* field, in order to prevent capability forgery.

2. $f_{stem}()$: is used to obtain a new *stem* number from the *ASN* and from either the *seed* or the current *stem* number of O.

To generate an *e-cap* C_1 for S_1, a number called $stem_1$ is first obtained by

$$f_{stem}(seed, ASN) = stem_1$$

After all direct information are put into C_1, the *check* field is computed based on $stem_1$:

$$f_{check}(S_1, stem_1, rights_1, lifetime_1, ACI_1) = check_1$$

Then, to generate an *e-cap* C_2 for S_2, a number called $stem_2$ is obtained from the $stem_1$ and ASN as

$$f_{stem}(stem_1, ASN) = stem_2$$

and the *check* field of C_2 is determined based on this new stem number by

$$f_{check}(S_2, stem_2, rights_2, lifetime_2, ACI_2) = check_2$$

Finally, the *check* field of C_3 for S_3 is determined by the following computations

$$f_{stem}(stem_2, ASN) = stem_3$$

$$f_{check}(S_3, stem_3, rights_3, lifetime_3, ACI_3) = check_3$$

Notice that all the C_i's contain the same ASN in their ACI fields.

Access restriction

When S_1 presents its C_1 for accessing O, the object server of O will first extract the ASN from its ACI field to compute $stem_1$. Then the same *check* field verification procedure is performed with the replacement of the *seed* number by $stem_1$ in verification. Since only C_1 contains a correct *check* field, only S_1 is allowed to access O. All other C_i's will not be verified as valid ones at this time, since their *check* fields are computed based upon different stem numbers. After the access of S_1, $stem_2$ is computed using $f_{check}()$ from $stem_1$ and ASN, and becomes the seed number in the next verification of the *check* field. Similarly, $stem_3$ will be computed to replace $stem_2$ and play the same role after the access of S_2.

It is quite obvious that each subject must follow the specified sequence in order to access O, because each C_i will not be treated as a valid one if it is not used at the right time. The number $stem_i$ is utilized as a *virtual seed* number of *O for this particular access sequence*. This number is modified immediately after the access of S_i, to make C_i just used invalid, and to make the object server only accept C_{i+1}, which allows no subjects but S_{i+1} to access O next.

Some applications may require that an access sequence repeat after the access of the last subject in the sequence. For example, a daily routine task needs a group of users to access a file in a fixed order everyday. This can be accomplished by storing additional information in the ACI field of the *e-cap* of the last subject, to advise the object server to reset the stem number after all the accesses in a sequence are finished.

Elimination of storing the *stem* number

The scheme for enforcing an access sequence described above is also storage efficient since only one *stem* number needs to be stored for each policy, and more favorably, it will not be produced until the access of the first subject in the sequence. Indeed, even the necessity of storing this number can be released at the cost of additional computation at each access. The access order of each subject in the sequence can also be specified in the ACI field of its *e-cap*. Thus, each stem number is generated from the *seed* number and the order information in the *e-cap* . With the example used above, $stem_2$ can be generated by calling $f_{check}()$ twice when C_2 is presented by S_2.

4 Capability Management

It has been demonstrated that an *e-cap* system is capable of enforcing a number of complex access control policies with an extensive use of the ACI field in an *e-cap*. We now discuss how *e-cap*'s can be propagated, revoked, and distributed.

4.1 Propagation of capabilities

Capability propagation is a mechanism to support granting of access rights from one subject to the other. Since an *e-cap* system is identity-based, a subject S_1 who wants to transfer its rights to another subject S_2 needs to explicitly make a request to the object server, along with its own *e-cap*, C_1. While it is a decision of the security policy whether a subject can transfer his rights to others, the object server can be configured to propagate C_2 to S_2 only when S_1 is the owner of the object (which can be indicated by a "owner" right), when a "transfer" right bit on C_1 is on, or after the object server checks the access control server to see if this right transfer complies with the security policy. Any of these alternatives can be specified in the ACI field when an *e-cap* is generated initially, so later the object server can take appropriate actions from this information in the *e-cap* after receiving a right propagation request.

The propagation tree suggested by the *ICAP* architecture [6] can also be incorporated in our *e-cap* system, yet in a distributed way. Whenever C_2 is propagated to S_2, the *id* of S_1, the subject which invokes the propagation (or just a pointer to it), can be embedded in the *ACI* field of C_2 to record where it is inherited from. A propagation tree can thus be built to keep track of all capability propagations, and the whole tree is actually distributed among different subjects in the system. When there is a need to know how access rights were propagated, we can upward trace the propagation tree by requesting each subject in the tracing path to present its capability in order to find its ancestor. For the general case in which only the owner of an object can transfer access rights, the depth of the tree is just two.

4.2 Revocation of capabilities

Revocation of capabilities is always a difficult problem in a capability-based system. This problem becomes more troublesome in modern distributed systems. When capabilities are manipulated in the user space, they cannot be revoked simply by a system-space mechanism like the back-pointers implemented in Multics [16]. In general, there exist three ways to revoke capabilities. First, an *expire* field can be used, on a per capability basis, to make a capability invalid after a pre-determined time period. The second method is to change the *seed* number associated with an object, which however invalidates all the capabilities generated based on this seed, and thus cannot support a selective revocation. The third way, suggested by Gong [6], is to maintain a *revocation list*, which stores all the revoked capabilities associated with an object. On every access, both the revocation list and the validity of the capability are checked in parallel. In order to avoid the inefficiency caused by searching a long revocation list, a *count* field can be associated with an object to determine how many capabilities have been issued for the object [19]. When the size of the revocation list becomes a significant fraction of the *count*, the object server just performs a permanent revocation by changing the *seed* of the object. However, re-issuing capabilities to subjects based on the new *seed* requires the object server to keep track of the propagation of all the capabilities, which may not be practical as well.

In our *e-cap* system, some capabilities can be efficently revoked by changing a virtual seed number associated with a security policy (as in the case of enforcing an access sequence) or by using the information in an *ACI* field (as in the case of implementing n-time tickets). Revocation of all the capabilities associated with a particular security policy can also be supported by maintaining a *policy revocation list*. When a security policy is not to be enforced any more, all the *e-cap*'s generated for that policy (they should have the same policy number in their *ACI* field) can be made useless by putting their policy number in the revocation list.

4.3 Distribution of capabilities

The methodology of distributing capabilities to subjects, adopted by most capability systems [1, 2, 6, 9, 11, 19], is to generate capabilities on demand. That is, a capability is not generated or distributed to a subject until it is needed. As a result, an object server often needs to check the access control server (usually after an object access) to determine when a capability should be generated and whom it is distribute to. The apparent disadvantage of this method is inefficiency, because too frequent checking with the access control server very possibly makes this centralized server a network and performance bottleneck when object servers are numerous.

Our *e-cap* system adopts a different methodology by that as many capabilities as possible are generated at once. When a security policy is to be enforced among several subjects, an object server obtains all the necessary information from the access control server to build all the capabilities at a time, and distributes them to the subjects before any actual access operation commences. Although the relations among the capabilities may become more complex (thus the cost of generating capabilities would be a little higher), the overhead of subsequent contact with the access control server can be diminished considerably. As shown previously, the object server also needs to possess mechanisms to process capabilities and to keep simple access control information for objects, which are usually kept by the access control server in other capability systems. However, the strategies of distributing access control information on capabilities earlier and of sharing access enforcement responsibilities with object servers are believed to be effective in balancing the storage requirement and enhancing the performance of the whole system.

5 Conclusion

We have proposed an innovative approach for enforcing complex access control policies in a capability-based distributed system. Within this approach, access control information is translated into the ACI's of capabilities by the access control server and distributed to subjects by object server. The object server is required to keep only simple capability processing rules and enforcement mechanisms. It has been demonstrated that many complex security policies can be enforced in a decentralized manner with efficiency in both time and storage. Our methodology of distributing all the capabilities for a security policy at once is also different from the conventional way of distributing capabilities on demand, and we believe it renders performance advantages over the latter since the communication overhead with the access control server is minimized.

References

[1] Paul Ammann, et al., "A Distributed Implementation of the Extended Schematic Protection Model," *Proceedings of the 7th Annual Computer Security Applications Conference*, San Antonio TX, December 1991, pp. 152 – 164.

[2] Jean Bacon, et al., "Extensible Access Control for a Hierarchy of Servers," *ACM Operating Systems Review*, Vol. 28, No. 3, July 1994, pp. 4 – 15.

[3] W. E. Boebert, "On the Inability of an Unmodified Capability Machine to Enforce The ∗-Property," *Proceedings of the 7th DoD/NBS Computer Security Conference*, September 1984, pp. 291 – 293.

[4] Jack B. Dennis and Earl C. Van Horn, "Programming Semantics for Multiprogrammed Computations," *Communications of the ACM*, Vol. 9, No. 3, March 1966, pp. 143 – 155.

[5] Li Gong, "On Security in Capability-Based Systems," *ACM Operating Systems Review*, Vol. 23, No. 2, April 1989, pp. 56 – 60.

[6] Li Gong, "A Secure Identity-Based Capability System," *Proceedings of the IEEE Symposium on Security and Privacy*, Oakland, CA, May 1989, pp. 56 – 63.

[7] Andrzej Goscinski, *Distributed Operating Systems: The Logical Design*, Addison-Wesley, Reading, MA, 1991.

[8] I-Lung Kao and Randy Chow, "Enforcement of Complex Security Policies with *BEAC*," *Proceedings of the 18th National Information Systems Security Conference*, Baltimore, MD, October 1995, pp. 1 – 8.

[9] Paul A. Karger and Andrew J. Herbert, "An Augmented Capability Architecture to Support Lattice Security and Traceability of Access," *Proceedings of the IEEE Symposium on Security and Privacy*, Oakland, CA, April 1984, pp. 2 – 12.

[10] Richard Y. Kain and Carl E. Landwehr, "On Access Checking in Capability-Based Systems," *Proceedings of the IEEE Symposium on Security and Privacy*, Oakland, CA, April 1986, pp. 95 – 100.

[11] Paul A. Karger, "Implementing Commercial Data Integrity with Secure Capabilities," *Proceedings of the IEEE Symposium on Security and Privacy*, Oakland, CA, April 1988, pp. 130 – 139.

[12] Bulter W. Lampson, "Protection," *Proceedings of the 5th Princeton Symposium on Information Sciences and Systems*, March 1971, pp. 437 – 443.

[13] H. M. Levy, *Capability-Based Computer Systems*, Digital Press, Bedford, MA, 1984.

[14] Dale A. Moir, "An Implementation of Access Control Using a Salient Feature of Primes," *Proceedings of the 7th Annual Computer Security Applications Conference*, San Antonio, TX, December 1991, pp. 298 – 322.

[15] National Computer Security Center, "Department of Defense Trusted Computer System Evaluation Criteria," DoD 5200.28-STD, December 1985.

[16] E. I. Organick, *The Multics System: An Examination of its Structure*, MIT Press, Cambridge, MA, 1972.

[17] R. F. Rashid, "Experiences with the Accent Network Operating System," in *Networking in Open Systems, Lecture Notes in Computer Sciences (248)*, pp 259 – 269.

[18] R. F. Rashid, "Mach: A New Foundation for Multiprocessor Systems Development," *COMPCON'87 — Digest of Papers*, pp 192 – 193.

[19] Ravi S. Sandhu and Gurpreet S. Suri, "A Distributed Implementation of The Transform Model," *Proceedings of the 14th National Computer Security Conference*, Washington, D.C., October 1991, pp. 177 – 187.

[20] Abraham Silberschatz and Peter B. Galvin, *Operating System Concepts*, 4th Edition, Addison-Wesley, Reading, MA, 1994.

[21] Lawrence Snyder, "Formal Model of Capability-Based Protection Systems," *IEEE Transactions on Computers*, Vol. C-30, No. 3, March 1981, pp. 172 – 181.

[22] Andrew S. Tanenbaum, et al., "Using Sparse Capabilities in a Distributed Operating Systems," *Proceedings of the 6th International Conference on Distributed Computing Systems*, Cambridge, MA, May 1986, pp. 558 – 563.

IGOR:
The Intelligence Guard for ONI Replication

R.W. Shore

The ISX Corporation
2000 North 15th Street, Suite 1000
Arlington, VA 22201
703/558-7800 (V), 703/558-7895 (F)
bshore@isx.com

Abstract: The Intelligence Guard for ONI Replication (IGOR) is a dual-host guard processor that allows database replication to occur across a security barrier with no person in the loop. IGOR works by accepting and validating SQL statements passed to it from a Sybase Replication Server (a product of Sybase, Inc.). A validated SQL statement flows across a serial line to the "other side" of the security barrier, where it is applied to the replicate database. IGOR's configuration files describe the SQL statements that are allowed to flow across the security barrier, including value checks that must be applied to validate the statements. Each of the two hosts associated with an IGOR installation is dedicated to processing SQL statements; only a limited number of UNIX users with well-defined roles are allowed to login to an IGOR host. IGOR has been accredited for a specific high-to-low installation. With different configuration files the same code can be used for other high-to-low situations, and with minor additions to the code IGOR would be appropriate for low-to-high situations as well.

This work was funded by the Office of Naval Intelligence, National Maritime Intelligence Center, 4251 Suitland Road, Washington DC 20395-5020. The Government point of contact is Mr. Al Poulin, 301/669-4000.

1. Introduction

One problem facing the Office of Naval Intelligence (ONI) is the dissemination of its analytical databases to customers at various security levels, in a timely and secure fashion. In the commercial world the technology of database replication is becoming one mechanism for keeping two (or more) copies of the same database synchronized automatically. Before this technology could be applied to ONI's problems, however, we had to develop a security guard that would allow the automated replication process to occur in a secure and controlled fashion.

This paper describes IGOR, the result of a nine-month effort by ONI to take advantage of the commercial replication software without compromising the security of ONI databases. The paper briefly discusses IGOR's operation and security features.

1.1. Glossary

GP
: Guard Processor. A role an IGOR host may play. The other role is the RSP.

IGOR
: The Intelligence Guard for ONI Replication. A GOTS product that allows a replication server to operate across a security barrier. Each IGOR installation consists of two hosts connected via a serial cable.

LTM
: Log Transfer Manager. Software that transfers changes from a master database into a Sybase replication server.

ONI
: The Office of Naval Intelligence, located in the National Maritime Intelligence Center (NMIC) in Suitland MD.

Replication Server
: A COTS product from Sybase that synchronizes a replicate database with a master by passing changes from the master to the replicate..

RSP
: Replicate-Side Processor. A role an IGOR host may play. The other role is the GP.

SQL

Structured Query Language. A near-standard syntax for expressing database changes.

1.2. Summary of IGOR's Operation

A replication server sends an SQL statement through a TCP/IP-based network to the host fulfilling the GP role. The GP verifies that the contents of the statement are in accordance with the security policy; specifically, the GP verifies that the statement mentions only the expected database, tables, and columns and that columns pass any value constraints given in the security policy. If the statement passes the checks, the GP passes the statement across a serial line to a second host fulfilling the RSP role. The RSP applies the statement to the target database via a second TCP/IP network and the appropriate, DBMS-specific protocol. The RSP returns a pass/fail status back through the serial line to the GP, which passes the status to the replication server.

IGOR's initial accreditation involved a high-to-low transfer, with the GP connected to an SCI network and the RSP connected to a non-US SECRET-level network; this is the mode discussed in the bulk of this paper. With relatively minor additional protections on the RSP side, IGOR appears accreditable for low-to-high operation. It is technically possible for each of a pair of IGOR hosts to fulfill both the GP and RSP roles, although there are no current plans to accredit IGOR in this mode.

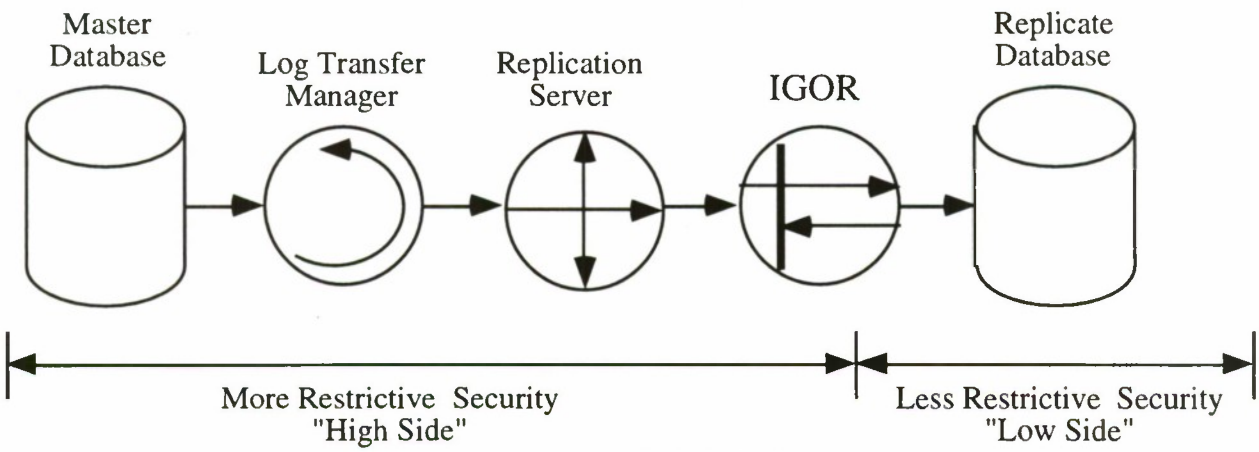


Exhibit 1 Basic IGOR Architecture

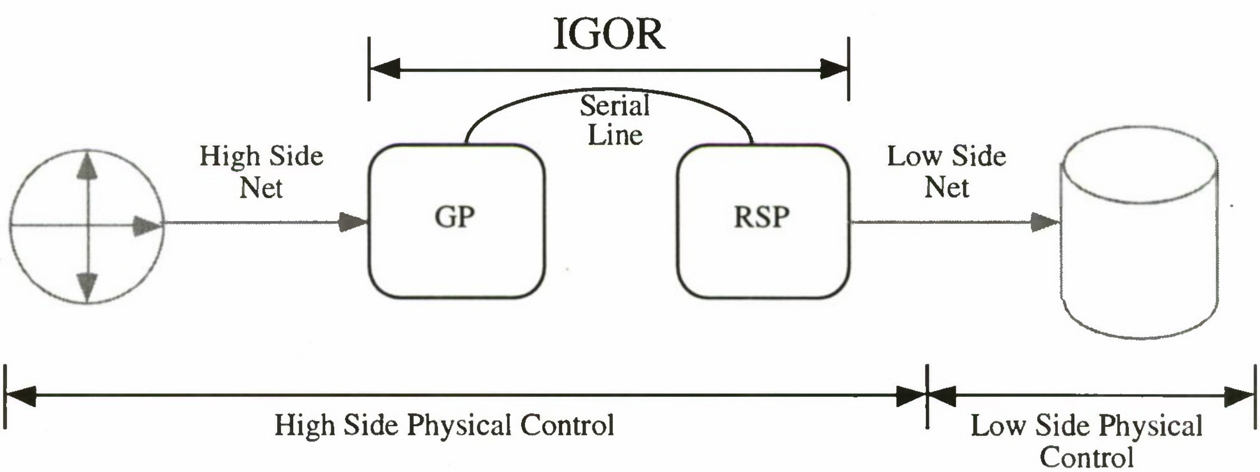


Exhibit 2 IGOR Hosts

2. IGOR's Operation

Exhibit 1 indicates the overall environment within which IGOR is to work. The following entities appear in Exhibit 1:

- Master Database: The source of changes are tables in a designated "master" database which may be managed by either Oracle or Sybase. Each change to a replicated table flows to a ...

- Log Transfer Manager (LTM): An LTM is an intermediary responsible for passing changes from a master database to a ...

- Replication Server: Within the replication server, changes to the master are queued and eventually distributed to one (or more) replicates. Exhibit 1 shows only one of an indefinite number of replicates, not all of which must involve IGOR. In the depicted situation the replication server passes each change to ...

- IGOR: IGOR verifies that the change being passed from the replication server is appropriate for this specific replicate. If the change is proper, IGOR simply applies it to the ...

- Replicate Database: This database contains a copy (perhaps a subset) of the master. The replicate can be managed by either Oracle or Sybase; the replicate's DBMS need not be the same as the master's.

The result of this process is that changes made to the master are also made to the replicate, in near real time and without a person in the loop.

Exhibit 2 shows a slightly expanded version of the IGOR instance appearing in Exhibit 1. As depicted in Exhibit 2, IGOR consists of two processors connected via a serial cable. The GP is connected to the same network as the master database; the RSP is connected to the same network as the replicate.

The replication server connects as a client to the GP and passes all changes to it, where a "change" takes the form of an SQL statement. For each row in the master that changes, the replication server emits one of the following SQL statements:

- `insert`: indicates that a new row has been added to a specific replicated table in the master.

- `delete`: indicates that a specific row has been removed from a specific table in the master.

- `update`: indicates that a specific row in a replicated table has changed.

Each SQL statement carries additional information to completely specify the change. For example, an `insert` statement includes the name of the replicated table, the names of all the columns, and the value associated with each column.

It should be noted that the replication process is a near real-time duplication of changes to the master. There is no filtering or consolidation of redundant changes at any step of the process. Suppose, for example, that a single transaction adds a row to a replicated table (`insert`), changes some values in that row (`update`), and then removes the row (`delete`). There is no net change to the master database. Even so, the replication process would faithfully reproduce this same sequence of SQL in the replicate database.

The replication server itself is a COTS product of Sybase Inc., which means that IGOR must live with the benefits and liabilities associated with COTS. Among the benefits is the fact that the replication server is a highly asynchronous operation.

- Changes move to the LTM only after the underlying database transaction has passed a commit point; the master's throughput is not seriously impacted by the replication process.

- The LTM uses either the transaction log (Sybase master) or trigger-maintained change-description tables (Oracle and other masters); while the LTM is processing a transaction, the update software running on the master is not blocked by the LTM's activities, nor are there any particular requirements levied on the master's maintenance software to support replication.

- The replication server accepts a change from an LTM by placing it into a disk-based "stable queue". Once the replication queues a change, the LTM is free to remove it from the master's tables or logs; barring a catastrophic disk failure, the replication server guarantees to deliver each queued change eventually.

- The replication server delivers changes to a replicate at the speed of the replicate, not the speed of the master. This is particularly important for IGOR, which is often limited by the speed of the serial line. Of course, the average change rate in the master cannot exceed the capabilities of IGOR; otherwise the replication server's queues will eventually fill.

As indicated by the preceding discussion, IGOR is the replicate database as far as the replication server is concerned. That is, the replication server logs into IGOR, passes SQL to IGOR, and gets success and error status returns, just as if IGOR were a Sybase SQL Server managing the replicated database directly.

From IGOR's perspective, of course, the situation is quite different. When it receives an SQL statement from the replication server, IGOR performs the following checks on it.

1. IGOR completely parses the statement. A statement that IGOR does not recognize as a proper `insert`, `delete`, or `update` generates an immediate error back to the replication server with nothing passed from the GP to the RSP.

2. IGOR verifies that name of the table appears in the statement and that the table is one IGOR expects to be replicated. If the table name is missing or incorrect, IGOR generates an error to the replication server and again passes nothing.

3. IGOR verifies that the statement names all columns and that each column is one IGOR expects. If the statement contains an unexpected column name or "anonymous" data (values not explicitly connected to a named column), IGOR generates an error back to the replication server without passing the SQL to the RSP.

4. For an `update` or `delete` statement, IGOR verifies that the `where` clause specifies a value for each primary key and that only primary-key columns appear in the `where` clause. If a key is missing or if the `where` clause contains an unexpected column, IGOR generates an immediate error back to the replication server without passing the SQL to the RSP.

5. For each columns with a value constraint (discussed subsequently), IGOR verifies that the value mentioned for that column is acceptable. If a column's value is bad, IGOR generates an immediate error to the replication server and passes nothing to the RSP.

6. If the table is subject to multi-table filtering, then IGOR runs the appropriate SQL and checks the return value. The need for multi-table filtering is discussed later in this section. If the change fails a multi-table filter, IGOR sends a "success" status to the replication server without passing the SQL to the RSP.

7. If the table is subject to full verification and the change is an `insert` or `update`, IGOR verifies that the row in question actually exists in the master database. If the row does not exist, IGOR sends a "success" status to the replication server without passing the SQL to the RSP.

If and only if the statement passes all of these checks, IGOR rebuilds the SQL from the representation generated by check #1 and passes the reconstituted statement to the RSP, which applies it to the replicate and passes a pass/fail status back. The GP sends the status to the replication server.

It should be noted that one of the reasons IGOR can reliably validate the statements passed from the replication server is that the server generates predictably-formatted SQL statements within a small subset of full SQL. For example, the server never generates a `select` statement, which is one of the more complex SQL statements to parse. Furthermore, the replication server's internal workings guarantee that each SQL statement describes a change to precisely one row of the master database. This is why check #4 above makes sense; each change must pick exactly one row by specifying a value for each primary-key column.

The checks that IGOR makes on each SQL statement can be divided into syntactic checks (#1–#4) and content checks (#5–#7). The syntactic checks verify that the statement is well-formed and mentions only the expected database, rows, and columns; these will be discussed no further here.

The value check (#5) ensures that the values in specified columns are in accordance with the security policy. IGOR allows the security policy to specify at most one wild-card expression (one UNIX regular expression) for each column; IGOR passes only values that match the expression. IGOR applies this check to the `values` clause of each `insert` statement to the `set` clause (not the `where` clause) of each `update`. For example, the security policy for IGOR's initial accreditation specified the following two restrictions:

- The value for column X in table T must be `M` or F. The associated regular expression is `'[MF]'` (the quotes are part of the expression).

- The value for column Y in table T must start with either M or N. The regular expression is `'[MN].*` (the quote is part of the expression).

Note that within the scope of check #5 IGOR examines a column only if it has a declared, specific UNIX regular expression. In particular, IGOR makes no attempt to do a generic "dirty value search" through all the columns being passed from the GP to the RSP; IGOR limits its checking to those columns constrained per the security policy.

Multi-table filtering (check #6) requires a bit more motivation. Consider, for example, a hypothetical high-side master database of aircraft locations (`ac_db`), and suppose that a low-side replicate needs to have only aircraft produced by a specific list of countries (`A`, `B`, `C`) passed to it. A typical database design for the master would put all the fixed information about aircraft (including the producing country `p_ctry`) in one table (`ac`) and the current location of the aircraft in another (`loc`); a key, such as a randomly-generated aircraft identifier (`ac_id`), indicates which rows in the location table are associated with which row in the aircraft table.

Now consider IGOR's dilemma when it is handed a change from `loc`. The security policy says that only aircraft produced by certain countries can be passed to the replicate, but a row from `loc` does not contain `p_ctry`. The only way that IGOR can decide whether or not to pass the change to the replicate is to consult `ac`, back in `ac_db`. The term "multi-table filtering" denotes that fact that IGOR needs information from other table(s) to determine the suitability of a particular row for the replicate.

In some cases it is possible to avoid the need for multi-table filtering by changing the database design. In the example above, for instance, the need for multi-table filtering disappears if we simply add `p_ctry` to `loc`. Database redesign is not always possible:

- Placing redundant data in tables is generally viewed as bad technical design and is often resisted by system analysts.

- Changing the structure of an existing production database and its maintenance software can be a long and expensive (thus undesirable) process.

When IGOR was being designed, it seemed prudent to implement multi-table filtering. There is a cost associated with multi-table filtering: for each change from a table subject to multi-table filtering, IGOR must validate the change by issuing a `select` statement back to the master database and checking the return value. This increases the transaction load on the master and may nearly double it if the table has a high rate of change.

It turns out that "multi-table" filtering can also be used to implement complex checks that cannot be handled by IGOR's simple column-by-column regular expressions. Suppose (to continue the hypothetical aircraft example) that IGOR is supposed to pass fighter aircraft produced in `A`, `B`, or `C` and transport aircraft produced in `X`, `Y`, and `Z`. The replication server can perform this sort of filtering (from a technical perspective, this constraint is an "or" of two "and"ed conditions), but IGOR cannot use its value checks to verify the replication server's filtering; IGOR's value-checking implementation does not support this sort of cross-column constraint. However, a "multi-table" filter that references only the `ac` table can be constructed so that IGOR will enforce this constraint. Again, however, the multi-table filter carries the penalty of a higher transaction load on the master.

Full validation (check #7) tells IGOR to ensure that each passed row actually exists in the master database. This check is very expensive in terms of the increased transaction load on the master and is not currently planned for use at ONI. It would be appropriate only when the master database is extremely sensitive and/or there appears to be a need for additional protection against uncontrolled, "rogue" programs attempting to use IGOR's facilities.

3. IGOR's Installation

The first requirement for an IGOR installation is a written security policy that specifies precisely what information is allowed to flow from the GP to the RSP and that is approved by (1) the owners of the information in the master database and (2) the appropriate security authorities. In technical terms, the security policy must be specific enough to specify a view of each replicated table. IGOR's basic job is to ensure that only the allowed view of each replicated table passes from the GP to the RSP. Some of the considerations associated with an IGOR installation appear in the subsequent sections.

3.1. Configuration Control

IGOR performs no queuing or other storage of SQL statements or database contents. All queuing occurs in the replication server; IGOR is a pass-through operation only. Thus except for deliberate maintenance activities (see Section 3.3 below), IGOR expects the content and location of most files to be static. To protect the configuration, the following features exist on both IGOR hosts.

1. IGOR runs with the keyboard disconnected and with no unnecessary peripherals (such as a CDROM drive) connected. This makes it more difficult to access the hardware console to perform a single-user boot.

2. After IGOR is installed, the superuser `root` is locked out. There is no way to gain interactive superuser status on an IGOR host; special IGOR setuid `root` applications provide limited `root` access to the UNIX logins on an IGOR host.

3. Most standard UNIX demons are not started. NFS and `sendmail`, for example, do not run. The only background process spawned by the `inetd` process is `telnet`; `ftp`, `finger`, and other such processes are not available.

4. Each time it starts, IGOR computes a file signature for critical configuration files and directories and compares the computed signature with a stored signature. If there is a mismatch, IGOR refuses to run.

Together, these features mean that it is difficult to change IGOR's configuration, and if an unexpected change does occur, IGOR shuts down the SQL transfer process. There is a back door to the IGOR host: a boot from a CDROM or other alternative media will allow an administrator to achieve single-user status and to unlock the `root` password so that interactive `root` access is possible. The absence of the keyboard and CDROM drive on the IGOR host during normal operation means that an alternative-device boot is a relatively complex and public process. Thus only a maintainer with proper authorization is likely to have access to the IGOR hardware for sufficient time to unlock `root`.

3.2. Access Control

As discussed in subsequent sections, IGOR includes two distinct access-control concepts: access via UNIX mechanisms and access via IGOR itself.

3.2.1. UNIX Access Control

Access to an IGOR host via UNIX mechanisms is limited by the following considerations.

- As mentioned previously, `root` is locked out. There is no way to achieve superuser status without an alternative-media boo.

- There are only two authorized userids on an IGOR host, conventionally called `igoradm` and `igorisso`. These userids have well-defined roles, as discussed in Section 3.3. All other userids in the password file are locked out at installation time.

- With most standard demons disabled, the only way one can access an IGOR host via UNIX is via `telnet` through the network.

Since `root` is locked out, there is no mechanism by which anyone can define a new userid. IGOR does include a setuid `root` module that allows a manager to clone a new administrator or ISSO, as discussed in Section 3.3. However, from a UNIX perspective a clone is identical to either `igoradm` or `igorisso` rather than being a separate and independent user.

3.2.2. IGOR Access Control

IGOR's GP is server software to which the replication server connects as a client. The GP has its own set of authorized userids and passwords, independent of the UNIX password file. IGOR recognizes two general classes of userids:

- An "incoming" userid is one that the replication server or other client uses to connect to IGOR.

- An outgoing userid is one that IGOR uses to connect to an external server. For example, IGOR needs an outgoing userid to connect to the replicate database and update it.

The passwords for these userids appear in IGOR's configuration files as encrypted values. For incoming userids, IGOR uses the same concept as UNIX to store passwords: the password field contains a value for which the clear-text password is the decryption key. Until an external user supplies the password to IGOR, the GP does not have the information it needs to decrypt the password field.

For the password for an outgoing userid, IGOR uses the fact that each incoming userid is associated at accreditation time with a single replicate database and set of verifications checks and hence with a fixed set of outgoing userids. IGOR stores the passwords for outgoing userids in encrypted format, using the clear-text *incoming* password as the decryption key. Thus IGOR needs the incoming password not only to validate access by a specific incoming userid but also to decrypt the necessary outgoing passwords.

Since IGOR uses the COTS OpenServer library from Sybase and expects connections from Sybase's replication server, any additional access control checks that IGOR might implement are constrained by the features provided by these two products. In particular, IGOR cannot reliably determine the host from which a connection is coming (the OpenServer does not provide this information) and cannot use any authentication scheme (such as a challenge-response sequence) beyond a simple password check (the replication server does not support any other scheme). This means that there is at least a theoretical possibility that an agent other than the replication server will attempt to connect to IGOR using the replication server's userid and password. IGOR implements the following obstacles to such an attack:

- The attack would have to come from the GP-side network. A user on the RSP network has no access whatsoever to the GP; even if the RSP is totally compromised the serial line between the GP and RSP uses a customized, IGOR-only protocol that provides no access to the GP's TCP/IP network.

- IGOR allows only one active connection for each incoming userid, and the replication server is generally connected to IGOR at all times. This limits IGOR's vulnerability window.

- If an attempt is made to open a second connection with an in-use userid, IGOR refuses the connection, shuts down the existing connection, and disables the userid. IGOR refuses all subsequent connections under that userid until the IGOR code restarts, either as a result of a reboot of the host or an administrative shutdown command to IGOR itself.

- The passwords associated with incoming and outgoing userids expire at an interval defined in IGOR's static configuration file, which is fixed at accreditation. This limits the length of time that an appropriated password will be valid.

- IGOR can be configured to verify that each row passed to the RSP is in fact in the master database. IGOR makes this check in addition to value checks and multi-table filtering. This option is very expensive in terms of the transaction load imposed on the master database, however, and is not currently used at ONI.

3.3. Maintenance

In general, IGOR maintenance follows a two-person rule: `igoradm` proposes and `igorisso` validates. Specific maintenance concepts include:

Database configuration: IGOR allows table names, column names, allowed values, and other parameters for SQL validation to change as the master database and security policy evolve. IGOR allows `igoradm` to propose a complete replacement for the database configuration file that describes the allowed SQL. `igorisso` must approve the replacement file (without change) before IGOR will actually use it. IGOR allows tables, columns, and so on, to change but does not allow a new database or database server to be added as either a master or replicate; server names appear in the static configuration file which is fixed at accreditation.

User configuration: IGOR provides a special application that `igoradm` and `igorisso` use to manage IGOR's incoming and outgoing userids. `igoradm` can add and remove entries from the user configuration file; `igorisso` can initialize and change passwords in existing entries. Note that a new userid cannot be employed until `igoradm` makes an entry in the configuration file and `igorisso` initializes the password.

Clone UNIX users: IGOR expects that there may be multiple individuals that can play either the administrative or ISSO roles and thus need UNIX passwords on an IGOR host. To help manage the UNIX passwords, IGOR provides a module that can clone either `igoradm` or `igorisso`. A clone for igorisso (for example) has a separate entry in the UNIX password file but runs under the same numeric userid as igorisso. A clone is simply an alternative password and is not an independent userid. igoradm (or any of its clones) can create a new clone; the clone is not usable until igorisso (or any of its clones) assigns an initial password. igoradm (or any of its closes) can remove a clone.

3.4. Alerting and Logging

IGOR was designed to run without a human operator and without human intervention most of the time. To keep its managers informed of various internal conditions, IGOR uses the UNIX mail system to send alerts to addresses outside of the IGOR hosts; although sendmail is disabled for incoming mail, IGOR can still send mail to external hosts. IGOR sends mail to an arbitrary number of addresses (specified in the database configuration file) whenever it starts up, whenever a serious error prevents IGOR from running, and whenever other "interesting" situations occur.

IGOR maintains a log of important events (UNIX login, IGOR login, and the like) and (on the GP) a complete list of all SQL statements sent to the RSP. A timed batch job (cron job) archives these logs, as well as other UNIX-maintained log files, to tape. The archive script Ssalvage checks for various error conditions during the archive run and alerts managers (via mail) when a tape needs to be replaced, the archive run fails, or other error conditions. Other than periodic replacement of a full archive tape, IGOR runs completely automatically, with no operator intervention required.

4. Status and Future Work

IGOR currently runs on two Sun Sparc IPX platforms running standard Solaris 2.4. Due to hardware limitations on the IPXs, IGOR's serial I/O is limited to 9600 baud. Two tables are being replicated from an ONI production database, one with approximately 20 attributes and the other with approximately 70. IGOR is handling about 15,000 inserts and 20,000 deletes per day, and could possibly handle as much as twice that load before the serial port bottleneck becomes critical. The CPU load on the system is low (generally under 20%); IGOR is definitely an I/O-bound process.

IGOR is built with Sybase's OpenServer product. It is multi-threaded; it can handle multiple connections and allows multiple tables to be replicated through a single connection. A design limit constrains each connection to involve a single source database and a single replicate database; the IGOR userid employed for the connection uniquely determines both the master and replicate databases, per the IGOR configuration files. Note that each IGOR installation is a guard between two specific security environments, one associated with the GP's network and the other associated with the RSP's network. Each distinct pair of security environments requires a separate IGOR installation.

During the week of 11-March-1996 this IGOR installation underwent accreditation tests by a team consisting of representatives from ONI-5 and DIA. The test uncovered no Category-I findings for IGOR itself (the only Cat-I finding involved accreditation for the master database). Of the two Category-II findings, one involved a minor code change and the other called for changes to IGOR documentation. There were several lower-category findings as well. All these findings have been resolved.

As mentioned in Section 1, the accredited IGOR installation operates in a high-to-low mode. Additional high-to-low IGOR installations would first require a written and approved security policy. The IGOR configuration files would next be constructed in accordance with this policy. The site would need to create installation-specific documentation, as an annex to the existing IGOR documentation, that describes the concept of operations and IGOR operational policies for the specific installation as well as a few security tests that depend on the structure of the master database. Finally, a security review or accreditation would be necessary to verify proper installation of the IGOR code on the new hosts and proper implementation of the security policy in the IGOR configuration files.

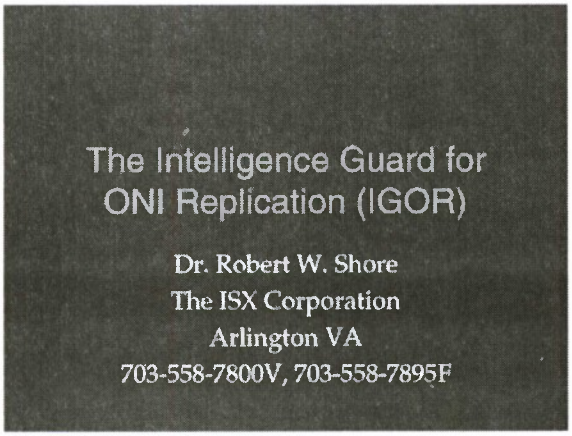
It is probable that low-to-high accreditation will require minor changes in the IGOR code. As implied by the other discussion in this paper, all of IGOR's validation activities currently take place on the GP. In the case of low-to-high replication, the GP roll is played by the IGOR host on the *low* network. Even though the GP itself is under the physical control of the high-side environment, there is at least a theoretical possibility that the GP could be compromised and all its protections removed. The following suggestions for IGOR changes to deal with low-to-high issues are proposals by the author; they are not sanctioned by ONI, nor have they been seriously discussed with any accreditation authority.

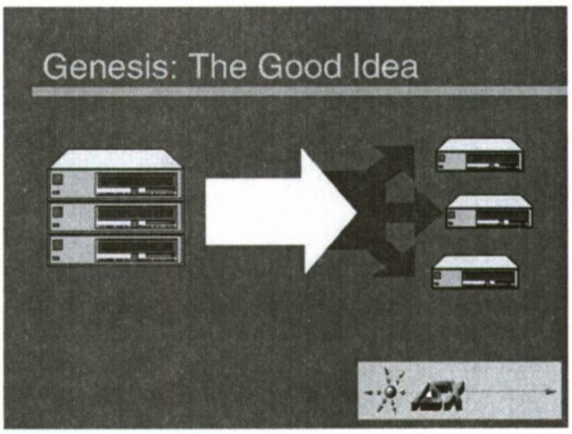
One protection that is clearly needed for low-to-high operation is a limit on the databases that the RSP can access. IGOR currently stores all access information on the GP side; the GP passes the name of the database and the userid/password to the RSP through the serial line. This approach is satisfactory for high-to-low operation; for low-to-high operation, the RSP should have its own table of allowed databases. This change is a fairly simple one in the RSP code. With the change in place, the RSP ignores the database identification passed from the GP. Instead, it uses the userid from the GP to look up the database and the real userid; the password from the GP is the decryption key for the real password. This approach not only limits the databases to which the RSP can connect, but also shields the real database name, userid, and password from the low-side IGOR maintainers. In addition, the DBMS privileges associated with the RSP's database userid can ensure that only the proper subset of the replicate database is visible to the RSP.

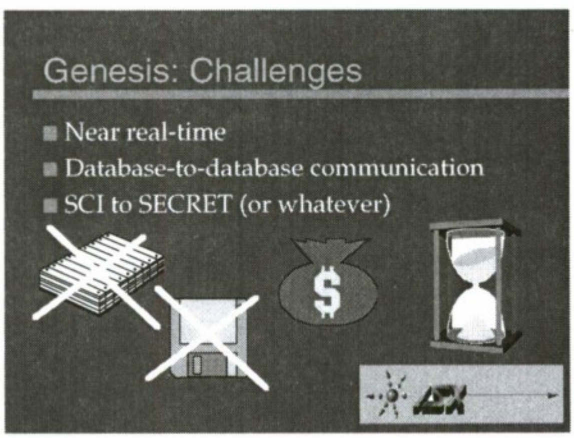
The RSP should also implement some SQL checks for low-to-high operation. For example, the RSP should verify that the SQL coming from the GP always has certain primary key fields with specific values; this check would ensure that data coming from the low side is properly marked and cannot be confused with similar data that originates on the high side. These checks are a fairly simple extension of the existing SQL parsing and checking capability already used on the GP side.

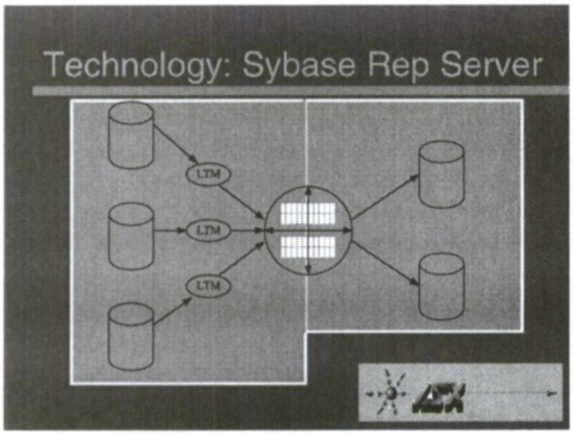
It was noted in Section 1 that the two IGOR hosts can theoretically play both the GP and RSP roles. That is, the situation might arise in which databases on the low-side databases need information from the high side and also high-side databases need information from the low side. The only code module in common between the low-to-high and high-to-low information flow in this situation is the serial-line handler, which has very limited functionality and thus can be verified to work properly without a great deal of work. At this point it seems that a dual-mode operation is feasible from a security perspective, although there are no plans to actually accredit any IGOR installation for this sort of operation.

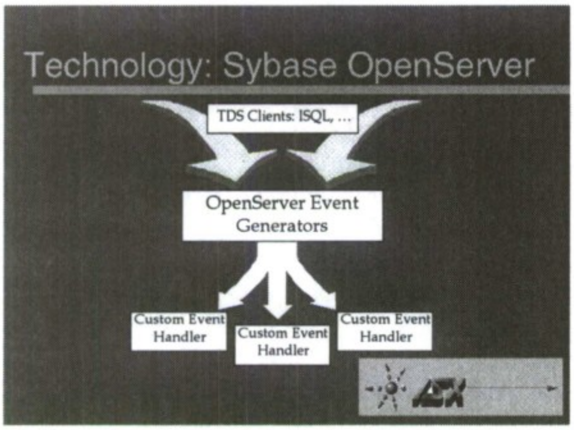
Finally, the reader will note that IGOR was built on standard Solaris. This situation exists mainly due to the lack of any approved product that would support Sybase's OpenServer and OpenClient libraries as well as Oracle's OCI library, but this situation will of course change over time. In a full multi-level environment with an approved, trusted operating system, IGOR's activities become that of a set of modules allowed to perform a specialized reclassification operation (write-down for high-to-low operation, write-up for low-to-high operation). In this environment, most of the SQL checks that IGOR currently performs would still be required. That is, IGOR would still have to verify that the SQL statements are appropriate for the RSP's security environment unless there are radical changes in the trusted versions of Oracle and Sybase and also in the replication server. Without such changes, most of the existing considerations, including multi-table filtering, would still apply. Although parts of IGOR's code would require modification for the new environment, much of it should port with little or no conceptual change.

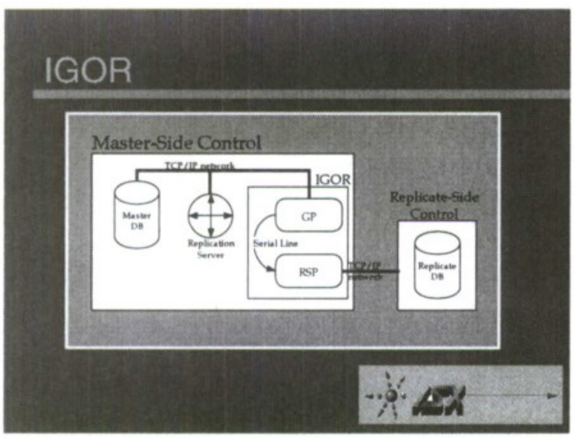

The Intelligence Guard for
ONI Replication (IGOR)
Dr. Robert W. Shore
The ISX Corporation
Arlington VA
703-558-7800V, 703-558-7895F


Genesis: The Good Idea

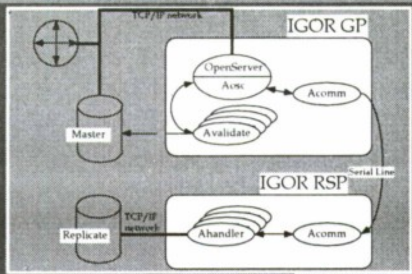

Genesis: Challenges
Near real-time
Database-to-database communication
SCI to SECRET (or whatever)


Technology: Sybase Rep Server
LTM
LTM
LTM


Technology: Sybase OpenServer
TDS Clients: ISQL, ...
OpenServer Event Generators
Custom Event Handler
Custom Event Handler
Custom Event Handler

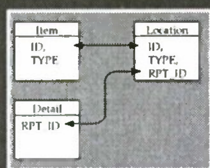

IGOR
Master-Side Control
TCP/IP network
IGOR
Master DB
Replication Server
GP
Serial Line
RSP
TCP/IP network
Replicate-Side Control
Replicate DB

IGOR Processing Modules



Complication

- Replicated table does not always include critical values
- Rep server cannot filter; IGOR cannot directly validate
- Requires "multi-table filtering" in IGOR



Features

- Fully multi-threaded
  - Multiple source databases per replication server
  - Multiple replication servers
- Near real-time delivery of changes
- Two-person rule support for all admin actions

Limitations

- Not designed to transport "bit blobs"
- Multi-table filtering can add a significant transaction load to the master database
- Single security level on GP side; single (but different) security level on RSP side

Key Concept: Security Policy

- Approved by data owners
- Specifies the view allowed in the replicate
- IGOR's job: verify that only information allowed by the policy passes

Status

- Code complete
  - Sun Solaris 2.4, gcc
  - Can probably be ported to other UNIXs
  - Built-in /dev/tty limited to 9600 baud
- Passed accreditation with minor findings

ETHICAL AND RESPONSIBLE BEHAVIOR FOR CHILDREN TO SENIOR CITIZENS IN THE INFORMATION AGE

Ms. Gale S. Warshawsky
Director and Programs Manager
Computer-Information Systems Security, Research & Practice
A Directorate Of:
ICICX -- International Community Interconnected Computing eXchange
General Secretariat • 415 Nahua Street • Suite 814
Honolulu, Hawaii 96815-2949 U.S.A.
E.mail for Ms. Warshawsky: msgale@gold.chem.hawaii.edu

Additional Contact:
Mr. Robert Mathews
Chairman - Steering Committee ICICX
E.mail for Mr. Mathews: mathews@gold.chem.hawaii.edu
E.mail for ICICX: icicx@maxwell.uhh.hawaii.edu

I. Communications Technology in the Information Age

In this age, communications technology has transformed our lives and changed the ways we communicate. For the purpose of this paper, communications technology includes: computers and information systems such as Local Area Networks (LANs), Wide Area Networks (WANs); fax machines and fax modems; personal communication services, such as, cellular and portable telephones, paging systems, voice mail, and the telecommunications infrastructure that enables populations to communicate via inter-networked systems.

Although this age is fascinating, it is far from golden. Users of computers and information systems must come to grips with the vulnerabilities inherent within inter-networked systems. Equally important is the issue that users need to learn how to apply the tools available on these systems in an ethical and responsible manner.

II. Double-Edged Sword

As a result of the development and use of computer and telecommunications technologies, our world has gotten smaller. For example, the Internet Telnet protocol allows users to visit different computers around the world. File Transfer Protocol (FTP) permits users to obtain files (including freeware and shareware) from other computers. Gopher and Veronica enable users to conduct information retrieval searches, and the Worldwide Web's (WWW) powerful search engines -- such as those available from Lycos®, Alta Vista®, and Yahoo® -- make it easy to conduct research around the globe from a computer in one's home, school or office.

Unfortunately the same tools that come to the aid of humanity and bring people together to work and play in the Information Age, can be used for a variety of unethical and criminal behaviors. We could be victims of white collar criminals using information systems.

Because users can be careless, unaware, and uneducated about information security, they often fail to protect their information. For example, hard disks do crash. It is not a question of whether a disk will crash, but when! It is therefore good business practice to perform backups. Many busy users however, get careless and don't back up their data. Many users, unaware of the problems that a computer virus may cause, do not install and use current programs that scan software for viruses. By not practicing good information security, users may become victims due to their own carelessness and/or lack of awareness and education.

Users may also become victimized by others' unethical or irresponsible behavior. Examples of such behavior include: phone fraud and electronic stalking or harassment, extortion, placing pirated software on a Bulletin Board System (BBS), or disseminating malicious software such as viruses, worms, Trojan Horses, or Logic Bombs. Victimizers include computer criminals and others who go astray or are lead astray by these criminals.

Readers of newspapers and publications on computer and information security are witness to headlines such as: "Clinton death threat is traced to Monte Vista High computer" [1], "Charges for Juvenile" [2], and "Pupils Cautioned for Card Fraud" [3]. The National Computer Security Association devoted an entire issue of its journal to ethics, with articles such as: "Totem and Taboo in Cyberspace," "A Question of Privacy," and "Why Hackers Do the Things They Do" [4].

The all-to-frequent articles about children committing computer crimes suggest that we as information security practitioners, must become pro-active in our efforts to change this situation. Inter-networked citizens must learn to practice responsible behaviors. As professionals, we must take the lead to ensure that computer users, our own and future generations, learn to use the inter-networked systems responsibly. Responsible and ethical behaviors need to be positively reinforced. Ethics in cyberspace needs to become normal, acceptable, and expected behavior.

III. Programs and Services

Many organizations that offer a variety of programs and services to the inter-networked global community are working toward that end. It is my honor to serve as a volunteer for one of them, ICICX -- International Community Interconnected Computing eXchange©. ICICX is a United States non-profit, charitable, scientific research and educational organization; its involvements are directed to the focused design, development, implementation and support of various Information Technology (IT)/Information Systems (IS) related services for the use of the inter-networked global community.

ICICX is composed of four directorates which shall focus to extend the benefit of their work to populations which ICICX recognizes as its constituency:

The first is CDTIES: Curriculum Development Technology Integration & Educational Services©. This is ICICX's Education Directorate, which will work hand-in-hand with educators, administrators, student leaders, and parents. It intends to define and develop curricula that encourage the integration of computers and information systems in a variety of educational environments, such as schools, home schools, libraries or other community centers.

The second directorate is IITCPD: Internetworked -- Information & Telematic Community Programs Development©. The effort of this ICICX Directorate focuses on working within a global community to survey and assess the telematic and informatic needs of a population. It is also the function of this directorate to pair programmatic, systemic, and specific service elements to the needs of populations.

Third we have ITSRDP: Informatic - Telematic Sciences Research Development & Practice©. This directorate focuses its attention on surveying, analysis, and reporting on topic areas such as information infrastructure: elements of telecommunications and information systems. ITSRDP is also a research and development arm of ICICX and will design and develop tools, services and materials technology. It will look at the ever changing environment and its impact on a society that has become dependent on using inter-connected systems.

The last, and most pertinent for my purpose today, is CISSRP: Computer-Information Systems Security, Research & Practice©. The programmatic areas of this directorate include several Policy Focus Areas; Awareness, Education, and Training for users of all ages, from children to senior citizens; Programs to reinforce positive and ethical behaviors within inter-networked systems; and Information about innovations in security for computer and information systems.

I volunteer my time away from my job at Lawrence Livermore National Laboratory (LLNL) to work with ICICX through its CISSRP Directorate. At LLNL, I am the Coordinator for Computer Security Training, Education and Awareness. In that position I design, develop and conduct training courses and produce awareness materials for LLNL personnel who need to protect information on LLNL computers and information systems in accordance with the U.S. Department of Energy's (DOE) Orders and LLNL policies. My work with ICICX/CISSRP will enable me to apply much of what I do on the job to a larger and more diverse community -- the inter-networked global community. As you well know, we face a Herculean task in raising awareness within this community and educating it about using information systems responsibly and securely.

ICICX, its board of directors, and directorates are a composite of a diversified virtual community. Since its members do not live near each other, there is a dependence on using communications and information technologies to interact, create, work, and recreate. We use a mixture of E.mail, real time inter-active sessions via the Internet, telephone, and faxes.

IV. Awareness, Education and Training

CISSRP plans to use awareness as a key ingredient to share concerns about the need to protect information and the importance of respecting authorship. This life-long process of awareness, education, and training needs to begin with young children and continue throughout their adult lives. The increasing numbers who communicate over the inter-networked systems need to be cognizant of basics such as these:

1. To Make a Good Password:
- Don't use personal information.
- Don't use dictionary words, in any language, spelled forward or backwards.
- Do combine letters and numbers to make a password that is easy for you to remember and hard for someone else to guess.

2. The Importance of Frequently Backing Up Information. Disks do crash.

3. How to Combat Viruses. Users need to understand:
What malicious software can do to a computer system and how to use current virus scanning software to detect and eradicate viruses.

4. Respect For Intellectual Property: Copyright, Trademark, Patent and Trade Secrets.

ICICX will use cartoon characters it has developed to share information on these and other information security areas with the global inter-networked community. This community is composed of a variety of generations. Senior citizens who have the desire to use computers need awareness, education, and training, just as much as children do.

V. Inter-networked Co-Learners

As participants in this community, we need to grow and expand our horizons together. Often, as we all know, learning about telematic and informatic technologies can be a frustrating experience. So we need to feel comfortable asking someone for help. Adults and children can be co-learners and co-educators, bridging generation gaps and helping one another to embrace the technologies of the Information Age.

To facilitate this process, ICICX has created a variety of cartoon characters who are user-friendly helpers. One such character was created to help users understand that the computer is a tool which does what it is requested to do. Frequently, users attempt to execute commands on an information system without really comprehending what the result of their actions might be. Many users get tangled and frustrated when attempting to use a computer or software programs and berate the system for not being a mind reader.

Another character we created struggles to understand how certain elements function within inter-networked systems. By providing engaging characters to facilitate learning through the ICICX Web Site and other educational materials, we hope to alleviate the anxiety felt by many new computer users of all ages.

CISSRP believes that learning is a life-long process that does not stop when one reaches retirement age. Senior citizens who retire at the age of fifty-five to sixty-five will expect to remain active as participants within society for many years past their retirement. Some senior citizen centers and libraries have computers for community members to use. With the decreased cost of hardware and software in the past few years, increased participation from senior citizens in the Information Age is an increasingly evident trend. Many purchase a computer system for use in their homes. In some cases, grown children bestow their senior citizen parents with the necessary tools to participate in the Information Age. Therefore, many of our senior citizens in the United States are embracing the Information Age along with their grandchildren.

Senior citizens have the ability communicate with members of their own age group and others in the larger global on-line community, thereby, narrowing the gap between the generations. SeniorNet through the use of the Worldwide Web (WWW) provides its participants the opportunity to enter into discussions and share a variety of subjects of particular interest to them. By accessing inter-networked systems, this segment of the population remains in communication with friends and colleagues, from the convenience of their homes and local communities.

CISSRP believes that our senior citizens have a wealth of information that could measurably enhance the lives of younger generations. Both seniors and youth could benefit from sharing their diversified knowledge bases with each other. CISSRP believes that seniors could serve as positive role models for our youth. Their maturity and ethical values may be shared with youngsters who are themselves beginning to expand their horizons.

CISSRP is equally concerned with the senior citizen community who are participating in the Information Age. Senior citizens may be unaware of the vulnerabilities that computer users can experience. CISSRP believes that this segment of our population is in as much need of information security awareness, education, and training as our youths are.

VI. Policies for an Inter-networked Global On-line Community

Every society follows rules and policies that enable it to co-exist. Drivers of motor vehicles are required to operate them according to federal, state, and local rules of the road. We vote for candidates to represent our interests. Some of us may involve ourselves in community organizations. In each of case, rules and policies exist which we as responsible citizens, agree to follow.

A Global On-line community has similar needs to our societal communities. Participants in this community must have policies that enable us to communicate responsibly and safely.

One of CISSRP's programmatic areas is Computer-Information Security Policies. Within this area, CISSRP will conduct research on a variety of policies and thus provide a place to share this information. CISSRP will collaborate with the ICICX Directorates that deal with education and community relations, to conduct and share research within communities that use inter-networked systems.

More broadly, areas of policy that interest CISSRP include: Values for the Inter-Networked Community, Ethics for the Inter-Networked Community, Essential Etiquette for the Inter-Networked Community, Guiding Principles, Responsible/Acceptable Usage Policies, Intellectual Property and Software/Hardware Piracy Issues.

VII. Resources

Within our society we have libraries, television, radio, newspapers, educational institutions, and on-line systems that provide us a wealth of information resources.

Analogously, CISSRP's Programmatic Area of Computer-Information Security Practice shall endeavor -- through Awareness, Education and Training -- to provide a central repository of pertinent resources for computer and information security practitioners. Within this programmatic area on the ICICX/CISSRP Web Site, you will find, "ICICX References," that will include information about:

- Videos -- Vendors and video titles with short descriptions of the videos and points of contact for ordering them.

- Organizations -- such as: American Society for Industrial Security (ASIS), Computer Ethics Institute (CEI), Computer Security Institute (CSI), Information Systems Security Association (ISSA), and National Computer Security Association (NCSA) [5].

- Materials -- where one may order Information Security Awareness materials, from outlets such as: National Computer Security Center (NCSC), CSI, NCSA, and Software Publishers Association (SPA) [6].

- Outreach Programs -- through ICICX and organizations involved in collaborative and cooperative agreements with it.

- Training -- Distance Learning through ICICX/CISSRP.

The cornerstone of ICICX/CISSRP's mission is a commitment to shape young populations, by a pro-active approach stressing Awareness, Education, and Training. Let us teach our youth now! Then, as they begin to use the inter-networked systems, they will use them responsibly and avoid irresponsible practices as they mature. CISSRP has in development a Children's Page.

It will feature ICICX's cartoon characters and offer creative, and amusing educational activities aimed at reinforcing ethical and responsible behaviors when children use computers and information systems.

I have been pro-active in my desire to impress upon young children the need to protect information on computers and to respect the intellectual property of others. Several years ago I developed, a local LLNL Computer Security Outreach Project that began when I volunteered as a result of LLNL Family Day Activities. The copyright to this work was eventually released by the Regents of the University of California, the U.S. DOE, and LLNL. The work was expanded upon and developed further during non-LLNL hours, resulting in the production of Chip & Friends™. This work was funded by and is copyrighted by the Atterbury Foundation, to which it was licensed. Chip & Friends was an effort to teach children in grades K-3 to be ethical and responsible users of computers. It consists of a video featuring puppets by Images In Motion; the video which is part of two 20 minute school presentations, is supplemented by a Teacher's Guide, a Parent's Guide, a Student Activity Book, a poster, and a small Chip plush hand puppet. The Chip & Friends materials are distributed by Computer Learning Foundation. [7]

VIII. Joining Forces -- Cooperative and Collaborative Efforts

Enlisting Chip & Friends to share information on ethical and responsible use of computers with young children was a good beginning. However, we need to continue and expand upon that effort. We need a myriad of people working together for the greater good of the inter-networked Global On-line Community. As Information Security Professionals, we need to share our expertise with the larger community. It in turn needs to embrace and foster information security, respect for intellectual property, ethical behavior, and responsible usage.

By joining forces we can accomplish a great deal. Everyone on earth should have access to the on-line inter-networked systems. By making such systems available to populations around the globe, we'll begin to tap an infinite potential for education.

As we continue to broaden our knowledge on using the communication technologies, let us at the same time, infuse our Global On-line Community with awareness, education, and training about responsible and ethical behavior.

ICICX/CISSRP urges all of us to share our expertise and knowledge within our local communities as well as with the larger on-line populations. We can join forces and cooperate. We can volunteer to address these essential subject areas in our local schools, at Parent Teacher Association (PTA) meetings, at community centers, at libraries, in our houses of worship, and in on-line discussion lists. Together, our forces joined, we can accomplish a great deal.

Here is just one example. A cooperative agreement between ICICX and the University of Hawaii at Hilo, was signed in February of 1996, when CSATI -- Center for Strategic Advancement for Telematics and Informatics© was formed. The Center was established to promote intellectual innovation in the development and deployment of the interconnected communications elements in all areas of telematic and informatic technology. CSATI's objectives include creating new relationships that will blend and synergize academic research, the business community, government, and industry.

We cordially invite persons and organizations wishing further information about ICICX and/or desiring to collaborate and cooperate with us to contact ICICX at:

International Community Interconnected Computing eXchange
Mr. Robert Mathews
Chairman - Steering Committee ICICX
General Secretariat • 415 Nahua Street • Suite 814
Honolulu, Hawaii 96815-2949 U.S.A.
E.mail for Mr. Mathews: mathews@gold.chem.hawaii.edu
E.mail for ICICX: icicx@maxwell.uhh.hawaii.edu
Telephone: 808.533.3969

REFERENCES

[1] "Clinton death threat is traced to Monte Vista High computer," San Ramon Valley Herald, January 19, 1996.

[2] "Charges for Juvenile," Computer Fraud and Security Bulletin, June 1995.

[3] "Pupils Cautioned for Card Fraud," Computer Fraud and Security Bulletin, July 1995.

[4] "Totem and Taboo in Cyberspace," "A Question of Privacy," and "Why Hackers Do the Things They Do.": Journal of the National Computer Security Association, June 1996.

[5] American Society for Industrial Security (ASIS)
1655 N. Fort Myer Drive, Suite 1200
Arlington, VA 22209-3198 USA
Telephone: 703.522.5800
FAX: 703.525.2694

Computer Ethics Institute (CEI)
PO Box 42672
Washington DC 20015
Telephone: 301.469.0615

Computer Security Institute (CSI)
600 Harrison Street
San Francisco, CA 94107 USA
Telephone: 415.905.2370
FAX: 415.905.2218

Information Systems Security Association (ISSA)
4350 DiPaolo Center
Glenview, IL 60025-5212
Telephone: 708.699.6441
FAX: 708.699.6369

National Computer Security Association (NCSA)
10 South Courthouse Avenue
Carlisle, PA 17013
Telephone: 717.258.1816
FAX: 717.243.8642

[6] National Computer Security Center (NCSC)
9800 Savage Road
Fort George G. Meade, MD, 20755-6000

Software Publishers Association (SPA)
1730 M St. NW Suite 700
Washington DC 20036
Telephone: 1.800.388.7478

[7] Computer Learning Foundation
PO Box 60007
Palo Alto, California 94306-0007
Telephone: 415.327.3347

PRIVACY RIGHTS IN A DIGITAL AGE

June, 1996

By William S. Galkin, Esq.
Law Office of William S. Galkin
10451 Mill Run Circle, Suite 400
Owings Mills, Maryland 21117
Telephone: 410-356-8853
Fax: 410-356-8804
E-mail: wgalkin@earthlink.net

A vast amount of information about each of us is being collected, compiled, sorted, transmitted, analyzed - and stored permanently. This information is gathered, manipulated and used without our consent - and usually without even our knowledge. This information includes credit histories, medical records, consumer purchases, email correspondence, and *much* more.

Methods of information gathering, though lawful, are becoming increasingly troubling. Here are two of many possible examples:

(1) There are Internet web sites designed for children where thousands of children pass through daily. Upon entrance to a site, a child is often asked to compete a questionnaire which requests information including age, background, interests, address, and phone number. Whenever before could such a vulnerable class such as children be freely approached, while alone, to disclose personal information?

(2) Millions of people are participating in discussion groups on the Internet known as newsgroups. The topics are all encompassing, including hobbies, politics, professions and personal relationships. The capability is now available whereby anyone can search these discussions, quickly, at no charge, from their desktop, and compile the comments made by any particular individual. The search will scan

millions of pages of information in seconds. This could be easily used by prospective employers, or by law enforcement agencies, for a variety of purposes, such as creating personality profiles.

Important privacy issues are arising in many different areas due to the increased use and availability of new technologies. This article will focus on three areas: employment, criminal investigations and encryption.

Source of Privacy Rights -

The "Right to Privacy" is a battle cry we often hear these days as we see our cherished realm of privacy being invaded by the onslaught of technology. However, legal scholars and the courts have had difficulty identifying the specific source of this right and defining its scope and application.

In 1928, the Supreme Court in *Olmstead v. United States* explained the right to privacy as the right to be "left alone." While many will agree with this description, it needs to be much further refined to be able to apply it in the many different situations where this right might arise.

Many believe that this right emanates from the Constitution. While it may, the U.S. Supreme Court has never expressly recognized a constitutionally-based right to privacy relating to collection and use of personal data, except as regards disclosure in criminal law proceedings. In 1965, in the case of *Griswald v. Connecticut*, the Supreme Court recognized a right to privacy relating to birth control counseling. This and subsequent cases identified the right to privacy relating to controlling an individual's life as relates to personal decisions. However, this does not provide a foundation for a right to privacy of personal information.

Others prefer to view the right to privacy as a property right, similar to the accepted corresponding property right found in the commercial context, namely, trade secrets. As a property right, the owner of this information would have the right not to disclose the information and to restrict others who received this information through a permitted disclosure from further disclosure in a manner that is not inconsistent with the "owner's" expressed instructions.

The comparison of trade secrets law with a right to privacy of personal information is difficult to take too far because the primary requirements for establishing a trade secret are not usually present in personal data. These requirements are (1) the secret information has value because it provides an economic advantage over competitors and (2) the information is actually secret, and the owner made reasonable efforts to maintain the secrecy of the information.

First, in the personal information context, the information itself has no value to the "owner," rather it is the disclosure of the information that has a negative value, though usually in a non-economic sense. Second, much of the information that people would like to keep secret is already lawfully in the possession of some company or government entity, and what we want is to stop further disclosure without our authorization.

The Employment Setting -

An employee, by the very nature of the employment relationship, must be subject to some level of monitoring by the employer. However, this monitoring has limits. Courts have held that it is a tortuous invasion of privacy for an employer to monitor employee telephone conversions. Similarly, mail carried through the U.S. postal service is granted a high level of protection.

However, much employee communication now takes place over private and public networks via email, or voice mail. These forms of communication are very different from telephone calls and letters. For example, after transmission and receipt, these communications are stored for an indefinite period of time on equipment under the exclusive control of the employer. Additionally, these communications can be examined without the knowledge of the communicators. As is often the case, the law has difficulty keeping pace with the issues raised by fast changing technology.

Electronic Communications Privacy Act - In the federal sphere, only the Electronic Communications Privacy Act of 1986 (ECPA) directly prohibits the interception of email transmissions. The ECPA prohibits the interception by (1) unauthorized individuals or (2) individuals working for a government entity, acting without a proper warrant. The ECPA is mostly concerned with the unauthorized access by employees or corporate competitors trying to find out valuable information. However, while there is no specific prohibition in the ECPA for an employer to monitor the email of employees, the ECPA does not specifically exempt employers.

The ECPA has several exceptions to the application of the prohibition of interception of electronic communications. The three most relevant to the workplace are (1) where one party consents, (2) where the provider of the communication service can monitor communications, and (3) where the monitoring is done in the ordinary course of business.

The first exception, consent, can be implied or actual. Several courts have placed a fairly high standard for establishing implied consent. For example one court held that "knowledge of the capability of monitoring alone cannot be considered implied consent." Accordingly, for an employer to ensure the presence of actual consent, it should prepare, with advice of counsel, a carefully worded email Policy Statement which explains the scope of employer monitoring. This

Policy Statement should be signed by the employees. One example of how this Policy Statement needs to be carefully written is that if it states that personal communications will be monitored only to determine whether there is business content in the communications, then this would probably not amount to consent to review the full text of personal communications. Additionally, notice that communications might be monitored may have a significantly different legal affect than a notice stating that communications will be monitored.

The second exemption is that the ECPA exempts from liability the person or entity providing the communication service. Where this service is provided by the employer, the ECPA has been interpreted as permitting the employers broad discretion to read and disclose the contents of email communications, without the employee's consent. However, employers should not rely on this exception, because it might not apply in all cases, such as to incoming (as opposed to internal email) if the email service is provided by a common carrier (e.g., America Online or MCI mail, which are not provided by the employer).

Under the third exception, courts will analyze whether the content of the interception was business or personal and allow the interception of only business-content communications.

State Laws in General - State tort laws are often viewed as the primary sources of protection for privacy of electronic communications. The most common tort that would apply is the tort of invasion of privacy. This tort occurs where "one who intentionally intrudes, physically or otherwise, upon the solitude or seclusion of another or his private affairs or concerns, is subject to liability to the other for invasion of his privacy, if the intrusion would be highly offensive to a reasonable person."

This tort does not require that personal information be actually acquired, disclosed or used. However, the intrusion must be intentional and highly offensive to a reasonable person. Additionally, there must be a reasonable expectation of privacy by the employee.

Employees often believe that their communications are private because they have a password which they can select and change independently or because they are communicating through outside common carriers. Cases have often turned upon whether this belief was reasonable given the fact that the employer had the ability all along to access the files, though the employees were not aware of this. In determining the outcome, courts will weigh the reasonableness of the employee's expectation of privacy against the business interest of the employer in monitoring the communication. However, it is important to emphasize that in the final analysis courts have traditionally held that legitimate business interests permit employers to intercept communications.

Law Enforcement -

The objectives of law enforcement and of personal privacy are on a collision course on the Information Highway. Law enforcement personnel desire access to as much information as possible to conduct their investigations. Individuals want to restrict access to personal information.

Recently, America Online under subpoena turned over personal email records relating to a criminal investigation where the murderer allegedly met the victim in an AOL chat room. AOL has been criticized by some for not challenging the subpoena. AOL's position is that if it receives a search warrant, it will comply. This case highlights the valid competing interests of both law enforcement and personal privacy. It is necessary to achieve a balance between these interests. How the Fourth Amendment to the U.S. Constitution is interpreted will play a crucial role in determining where this balance is reached.

The 4th Amendment prohibits government agents from conducting unreasonable searches and seizures. The Supreme Court has defined a seizure of property as a "meaningful interference with an individual's possessory interest in that property." The concept of seizure of information differs dramatically from seizure of tangible property. Seizure of tangible property means that the owner has been deprived of the use and possession of the property. Whereas, when information is "seized" the owner may still have possession of the information. It is just that the information has been copied and is now also in the hands of someone else.

It could be argued that under the Fourth Amendment no seizure occurs when digital information is merely copied. However, applying the analysis used to prohibit wiretapping (which has been defined as a seizure), seizure of information would also fall within the constitutional definition of seizure. In the information context, "seizure" should be interpreted as meaning being deprived of the ability to control the disclosure and dissemination of the information. This ability to control is the value of the possessory interest of information.

The application of the term "search" in the digital environment is more complicated. An unlawful search requires as a prerequisite that (1) subjectively, the person in possession of the item searched had an actual expectation of privacy and (2) objectively, the person had an expectation of privacy. The subjective expectation of privacy element has been criticized, because in theory, it would be very easy for the government to eliminate any expectation of privacy by announcing that it will perform broad searches. However, in practice, the Supreme Court has focused on the objective requirement.

On one end of the spectrum is data resident in a stand-alone computer. Here, there is certainly an objective expectation of privacy. On the other end of the spectrum lie the vast open areas of the Internet, such as web pages and newsgroups to which there can be no objective expectation of privacy. Accordingly, law enforcement agents are free to roam through these open areas, assemble records on who is participating in which groups, and what they are saying. The middle ground is where the legal battles will be fought. This will primarily involve information that is in the possession of a third party, and is not readily accessible to the public.

Under traditional Constitutional analysis, where information is disclosed to a third party, the expectation of privacy is abandoned. For example, most state laws, and the federal Constitution, permit wiretapping if one party to the conversation consents. However, the scope of the abandonment will usually only apply to the amount of information needed by the recipient.

For example, the telephone numbers you dial are disclosed to the phone company in order that the phone company can perform its service. Thereby, a person abandons the expectation regarding the number dialed. However, even though the content of telephone conversations is also given over to the phone company, this content is not needed for the phone company to perform its service. Therefore, the content of phone conversations retains the expectation of privacy.

By analogy, this would also apply to email messages maintained on a service provider's equipment. Information such as the senders' and recipients' addresses, the file sizes and times of transmissions are not private. But the content of the messages would be.

In the workplace, an employer is not permitted to consent to a search of personal areas of an employee. For example, a desk draw that contains personal correspondence. By accepted convention, this is a private area.

Private network directories which require a password to enter would probably also retain an expectation of privacy. However, in each case, a court will look at specific corporate policies to determine whether there is an objective expectation of privacy or whether the employee was informed that the employer may at any time without notice enter these pass-worded directories.

Along these lines, since a court wants to determine the objective expectation of privacy, an agreement that an employer will not consent to a search would have no effect. What would be needed is an agreement that the employer will not access these private areas, which deprives the employer of the right to consent.

When determining the objective expectation privacy, courts will have to balance the value of the particular privacy interest claimed against the level of the law enforcement interest.

Encryption -

Cryptography is the ancient art of concealing the content of a message by scrambling the text. Historically, it was used for communicating military secrets. Now, the secrets might be commercial, personal, political or criminal, and communicated over the Internet.

A would-be reader of an encrypted message must have a "key" to descramble the message. Encryption software, the modern method of encryption, uses a mathematical algorithm to scramble a message.

There are two primary forms of encryption software: single-key systems and two-key systems. In a single-key system, the data is encrypted and decrypted using the same key. The weaknesses of the single-key system are that the key is not completely secret because both the sender and the receiver must have the key. Additionally, at some point prior to the first encrypted communication, the key itself must be communicated in an manner that does not use the same encryption method.

A two-key system, also known as a public key system does not have these weaknesses. This system uses two keys, one private and the other public. The public key is given out freely and will encrypt a message. However, only the private key, which does not need to be communicated to anyone, can decrypt the message. It is practically impossible to determine the private key from an examination of the public key.

Encryption Regulations - The Bureau of Export Administration (BXA), under the U.S. Commerce Department, controls licensing for most exports from the U.S. However, the BXA is excluded from controlling items listed on the U.S. Munitions List. The Munitions List designations are made by the State Department with the concurrence of the Defense Department, and are contained in the International Traffic in Arms Regulations (ITAR).

The Munitions List includes things like grenades, torpedoes, and ballistic missiles. The list also includes "cryptographic (including key management) systems, equipment, assemblies, modules, integrated circuits, components or software with the capability of maintaining secrecy or confidentiality of information or information systems."

The State Department relies on the National Security Agency's (NSA) expertise when deciding what encryption programs to include on the Munitions List. The NSA, a member of the U.S. Intelligence Community under the Defense

Department, is responsible for decoding the signals of foreign governments and collecting information for counterintelligence purposes. The NSA review process is classified and not available to the public. However, generally, if the NSA cannot relatively easily break an encryption algorithm, it will not approve it for export.

A violation of the export restrictions on encryption can result in a maximum criminal penalty of $1 million and 10 years in prison or a maximum civil penalty of $500,000 and a three-year export ban.

There are no restrictions on encryption systems contained in software marketed solely in the U.S. Most other countries do not restrict export of encryption software. However, in France, the private use of cryptology is not permitted, unless the government is provided with the private key.

Effectiveness of the Law - It is questionable how well the current law achieves its objectives. The encryption export restrictions are intended to protect the national security of the U.S. However, since much sophisticated encryption software is now being developed out of the U.S., it is unclear how important these restrictions remain. Additionally, national security is threatened from both internal as well as external sources. Therefore, since there are currently no restrictions on the use, development or distribution of encryption software in the U.S., these restrictions play little role in guarding against internal threats.

The law also produces some strange results. Encryption software can be imported into the U.S., but the same software cannot later be taken out of the U.S. A U.S. citizen can develop sophisticated encryption software abroad and have it marketed internationally, but cannot do the same if the development occurs in the U.S. The State Department has ruled that a book on applied cryptology, which contains source code for strong encryption algorithms may be exported, but the verbatim text of the source code when on a computer disk cannot.

Changes in the Law - The government has been moving in two directions at once. While there has been some lifting of the restrictions on the export of encryption software, there have also been developments indicating that the government desires to gain a "back door" to allow law enforcement officials the ultimate ability to access any encrypted message.

One example of the lifting of restrictions was in 1992, when mass marketed software with "light" encryption was made subject to an expedited 15-day or 7-day review by the State Department. This increased the likelihood that export licenses would be granted for such software. Additionally, effective this year, under certain circumstances, a U.S. citizen may temporarily take encryption software abroad for personal use.

However, at the same time the export restrictions are being lightened, several government initiatives have attempted to grant the government skeleton keys to access encrypted messages, such as the 1993 Clipper Chip initiative and the Escrowed Encryption Standard mandated for the federal government. Both of these developments seek to provide the government with the ability to access private keys. Furthermore, there has even been mention of seeking to criminalize the use of encryption in the U.S., unless private keys are escrowed with the government, as is currently the law in France.

Most recently, on March 5, 1996, Sen Patrick Leahy (D-VT) introduced the Encrypted Communications Privacy Act of 1996 (S. 1587). If it becomes law, the Act would (1) remove export restrictions for "generally available" encryption software, (2) shift authority for export decisions from the State Department and NSA to the Commerce Department, (3) criminalize the use of encryption to obstruct the investigation of a felony, and (4) regulate disclosure of encryption keys by key escrow agents.

The Act would greatly loosen the restrictions on exporting encryption software. However, it would still probably be up to the NSA to determine what software is "generally available." Furthermore, since the exclusion will be limited to encryption software that is generally available, U.S. companies will always be lagging behind foreign competitors, because U.S. companies will not be permitted to take the lead in the international marketing of cutting edge encryption products.

Lastly, some have expressed concern over two features of the Act. One is that the Act sets forth the first instance in the U.S. of specifically criminalizing the use of encryption. And second, if private key escrow is intended to remain voluntary why is so much of the Act devoted to escrow issues?

The encryption debate has a long way to go and reflects a fundamental struggle between ensuring personal freedom while providing the government with the means of maintaining a safe society.

Conclusion -

Much of the law of privacy turns on the reasonable expectation of privacy. When evaluating different situations, it is important to keep in mind that the law in this area is a moving target, as expressed by Professor David Post of Georgetown University Law Center (in *The American Lawyer*, October 1995) "until we have all spent more time in this new electronic environment, who can say what our expectations really are --let alone whether they are reasonable?"

TRUST TECHNOLOGY ASSESSMENT PROGRAM

PANEL MEMBERS:
Tom Anderson, NSA, Chairperson, TEAnderson@Dockmaster.ncsc.mil
Pat Toth, NIST, Toth@csmes.ncsl.nist.gov
TTAP Working Group Members, TTAP@csmes.ncsl.nist.gov

This panel will focus on the progress of the Trust Technology Assessment Program (TTAP) initiative including the lessons learned from the prototype effort to validate the process, procedures, and documentation to support the program in a commercial environment. Additionally, the panel will provided feedback to the community on the outcome of the public workshop on the TTAP held in September 1996. The panel will also provide insight into future activities associated with product testing and evaluation currently under discussion within NIST and NSA.

The TTAP is a joint National Security Agency (NSA) and National Institute of Standards and Technology (NIST) effort to commercialize lower level of trust evaluation of commercial-off-the-shelf (COTS) products. Under the Auspices of the National Voluntary Laboratory Accreditation Program (NVLAP), TTAP will establish, approve, and oversee commercial evaluation laboratories focusing initially on products with features and assurances characterized by the Trusted Computer Systems Evaluation Criteria (TCSEC) Class B1 and lower levels of trust. Vendors desiring a level of trust evaluation will contract with an accredited laboratory and pay a fee for their product's evaluation.

Panel: Alternative Assurance: There's Gotta Be a Better Way!

Abstract: Traditional methods for ensuring that policies are enforced by Information Technology have proven slow and ill-matched for many of today's needs. This panel is designed to highlight the events at the June '96 Workshop on Information Technology Assurance and Trustworthiness (WITAT '96) towards evolving practical solutions for business and industry in need of confidence in their information systems. This panel will explore the available alternative assurance approaches and discuss their use for today's expanded and demanding assurance needs. Areas of assurance explored include, assurance predictors, system analysis and operational assurance, and impact mitigation.

Douglas J. Landoll
Arca Systems, Inc.
(703) 734-5611
(703) 790-0385 Fax
landoll@arca.com

ALTERNATIVE ASSURANCE: THERE'S GOTTA BE A BETTER WAY!

This panel is designed to highlight the events at the Workshop on Information Technology Assurance and Trustworthiness (WITAT '96), held on Sept. 3- 5, 1996. This workshop is intended as an initial step towards evolving practical solutions for business and industry in need of confidence in their information systems. The focus of this year's WITAT is to determine the merits of alternative assurance approaches and to create a strategy for developing the promising areas. Issues about these alternative assurances will be discussed between audience members and panelists. Additionally, results of the workshop and plans for developing promising assurance methods will be presented. The panelists are industry experts who will be chosen as subgroup chairs during WITAT '96.

WITAT '96 - In 1994, the Aerospace Computer Security Associates (ACSA) and the National Institute of Standards and Technology, responding to a perceived growing need in the community, organized and sponsored the Invitational Workshop on Information Technology Assurance and Trustworthiness (IWITAT). The success of this workshop led to WITAT '95 and now the planning for WITAT '96.PANEL DESCRIPTION

Panel Introduction (10 minutes)
Doug Landoll (WITAT '96 Chairman) Arca Systems, Inc.

Traditional methods for ensuring that policies are enforced by Information Technology have proven slow and ill-matched for many of today's needs. This panelist will establish a framework for the remainder of the panel to explore the available alternative assurance approaches and discuss their use for today's expanded and demanding assurance needs.

Assurance Predictors (20 min. -15 presentation, 5 questions)
Mr. John J. Adams, NSA

Mr. Adams has focused his work at NSA for the past 3 years on alternative assurance methods. Two projects of note are the SSE-CMM and the TCMM. He participated in WITAT '96 and will report on the results of the workshop's discussion on Assurance Predictors.

Can assurance in an information system be gained from looking at the capability of the organization or individuals involved in develop/integrating/maintaining/operating the system? There are many methods that provide information about organizational or individual capability. What assurance do these methods provide? WITAT '96 discussed various methods that indicate an organization's or individual's capabilities in an attempt to answer the above questions. The methods to be discussed include: Capability Maturity Models (CMMs), the Generally Accepted System Security Practices (GSSP), International Information System Security Certification Consortium (ISC2), ISO 9000 series, Past Performance and Trusted Software Development Methodology (TSDM).

System Analysis & Operational Assurance (20 min. - 15 presentation, 5 questions)
(System Analysis & Operational Assurance Subgroup Chair)

System Analysis: The most direct way to achieve assurance in an information system is to analyze it directly. This panelist will discuss traditional authoritative methods such as TPEP and ITSEM and the acceptance of less authoritative independent testing.

Operational Assurance: Product and system assurance is only one ingredient involved in gaining confidence in an operation. Operational assurance depends not only on the information technology, but also on the people, environment, and processes involved. Even if information technology was 100% free of flaws, people would have to install, configure, and use it correctly to be secure. A panel will discuss the available methods for gaining operational assurance. The methods studied included: setting policy, risk assessment, background checks, configuration management, training, monitoring, and incident response.

Impact Mitigation (20 min. -15 presentation, 5 questions)
(Impact Mitigation Subgroup Chair)

Other known assurance techniques focus on reducing the vulnerabilities of the information system. These new types of assurance are not related to avoiding vulnerabilities of the system at all, but instead seek to mitigate the impact of defects usually in the form of software fixes or monetary reimbursement. This panelist will discuss several impact reduction assurance methods including warrantees, insurance, and legal liability.

Determining the Appropriate Mix (20 min. -15 presentation, 5 questions)
(Determining Assurance Mix Subgroup Chair)

What is the right mix of assurance approaches for your organization? This panelist will discuss the most effective combinations of assurance approaches for commercial and government systems, depending upon factors such as environment, reliance on technology, value of reputation, impact of security breaches, and connectivity needs. Different ways of composing assurance approaches will be presented including: assurance arguments, trade-offs, and criteria.

CERTIFICATION AND ACCREDITATION - PROCESSES AND LESSONS LEARNED

Chair: Mr. Jack Eller, DISA, CISS (ISBEC)

Panelists: Paul Wisniewski, National Security Agency

Candice Stark, Computer Sciences Corporation

Ray Snouffer, National Institute of Standards and Technology

Barry C. Stauffer, CORBETT Technologies, Inc.

Panel Summary

Mr. Jack Eller, DISA
CISS (ISBEC)
701 South Courthouse Rd.
Arlington, VA 22204-4507
(703) 681-7929, ellerj@ncr.disa.mil

On August 19, 1992 the Office of Assistant Secretary of Defense directed the Defense Information Systems Agency (DISA) Center for Information Systems Security (CISS) to formulate a standard DoD process for security certification and accreditation. CISS formed a working group, consisting of Service and Agency representatives. The working group evaluated ten existing processes, but found none which could be adopted Department of Defense (DoD)-wide. As a result, the working group developed the DoD Information Technology Security Certification and Accreditation Process (DITSCAP). A uniform process across DoD, DITSCAP applies to accreditation of both strategic and tactical systems, as well as stand-alone information systems or networks. DITSCAP capitalized on approved security techniques, software, and procedures to reduce the complexity and overall cost of the accreditation process. The DITSCAP integrates security directly into the system life cycle and is designed so that it can be applied uniformly across DoD. The DITSCAP defines a process which standardizes all activities leading to a successful accreditation, thereby minimizing the risks associated with nonstandard security implementations across shared Defense Information Infrastructure (DII) and end systems. The DITSCAP has been designed to support the requirements of Office of Management and Budget Circular A-130.

In contrast to the prevailing system based accreditation processes, the DITSCAP is focused on the infrastructure and views systems and networks as components of the infrastructure. The view of the DITSCAP, therefore, differs from such documents as the National Computer Security Center (NCSC) Certification and Accreditation Process Handbook for Certifiers (NCSC-TG-031). CISS and the NCSC have agreed that for the near term, NCSC-TG-031 provides sound guidelines. DITSCAP provides the midterm and long term infrastructure-centric approach to the security certification and accreditation of systems and networks. These two processes have been harmonized to reflect the transition to the DITSCAP. Both terminology and structural parallels will facilitate a smooth transition between these two processes.

Our panelists today will present an overview of the elements and approval status of the Certification and Accreditation Process Handbook for Certifiers, an the Certification and Accreditation Process Handbook for Accreditors. Following these presentations we have two presentations which will discuss some lessons learned in applying each of the two processes.

THE CERTIFICATION AND ACCREDITATION PROCESS HANDBOOK FOR CERTIFIERS

Paul Wisniewski
National Security Agency
Office of Commercial Programs and Enabling Technologies
9800 Savage Road
Ft. Meade, MD 20755-6740
(410) 859-6281, pawoeck@radium.ncsc.mil

The National Computer Security Center is publishing the *Certification and Accreditation Process Handbook for Certifiers* as part of the "Rainbow Series" of documents. This document continues the series on certification and accreditation (C&A) and provides the certifier and accreditor with a structured process to perform a C&A of a system. It should be viewed as guidance in determining the amount of effort and the resources necessary to certify and accredit a system. As technology that supports the infrastructure of automated systems becomes more sophisticated, the C&A process will, no doubt, require new or additional guidance. However, this document provides the necessary certification and accreditation guidance for now and into the near future.

The terminology and structure in the *Certification and Accreditation Process Handbook for Certifiers* has been harmonized with the *DoD Information Technology Security Certification and Accreditation Process (DITSCAP)*. Thus DoD elements may use this document in support of their C&A requirements. However, this document is not DoD specific. The C&A process described is consistent with the earlier guideline, *Introduction to Certification and Accreditation.* Non DoD agencies and organizations should have few problems in seeing the parallels and using this latest document to support their C&A programs.

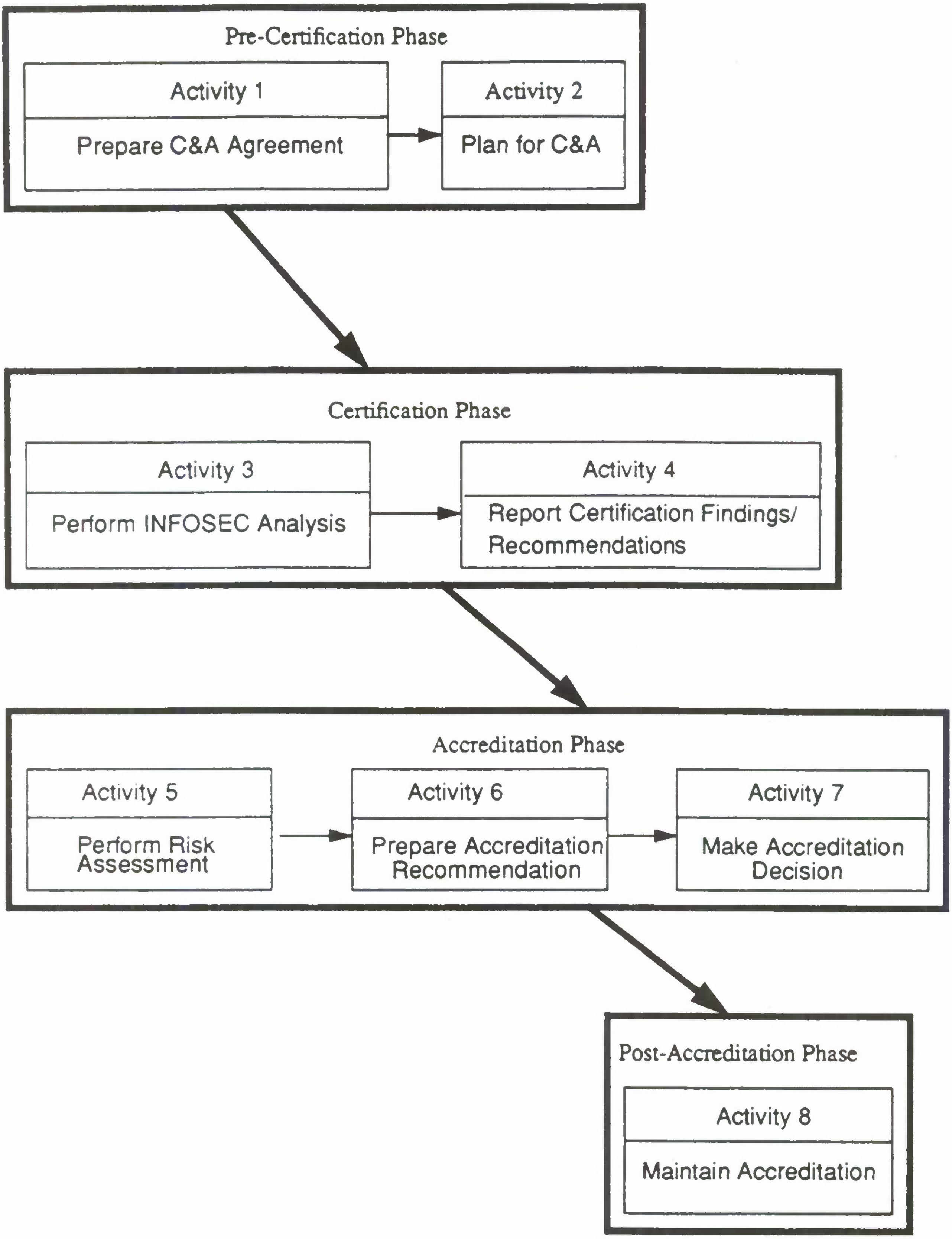
The purpose of this handbook is to establish a standard approach for performing C&A on systems regardless of the acquisition strategy or life-cycle status. This handbook provides guidance about the C&A process based on the degrees of assurance required and other factors related to a system. Assurance is a measure of confidence that the security features, attributes, and functions enforce the security policy. Assurance refers to the claims for evidence for believing the correctness, effectiveness, and workmanship of the security service or mechanism. Certification verifies and validates the security assurance fora system associated with an environment. Accreditation evaluates whether the operational impacts associated with any residual system weaknesses are tolerable or unacceptable. The degrees of assurance assumed by a development team, certification team, or Accreditor about a system reflect the confidence that the system is able to enforce its security policy correctly during use and in face of attacks.

The C&A process allows the DAA, Program Manager, and User representative to tailor the certification efforts to the particular system mission, threats, environment, degrees of assurance, and criticality of the system, as necessary, as long as they comply with network connection rules. With a standard approach established, reuse of both the technical and nontechnical analyses from the certification effort , for recertification or certification of a similar system, might be possible. The C&A process should encourage and preserve commonality in understanding, be consistent in application, be open to evolution and growth, employ feedback, and be applied continuously. This process should be scalable to the size of the system, repeatable, and predicable.

STANDARDS IN CERTIFICATION AND ACCREDITATION

Candice Stark
Computer Sciences Corporation
7471 Candelwood Road
Hanover, MD 21076
(410) 684-6329

This presentation will address the why, who, what, how and where of C&A standards. The speaker will expand on the latest in the Rainbow series C&A sub-series, the Accreditor's Guide. Ms. Stark was initially immersed in C&A while at NSA. While there she was intimately involved with the creation/editing of the three C&A documents in the Rainbow series. Now at CSC, she is still involved in C&A issues for the DoD.


Pre-Certification Phase
Activity 1
Prepare C&A Agreement
Activity 2
Plan for C&A
Certification Phase
Activity 3
Perform INFOSEC Analysis
Activity 4
Report Certification Findings/ Recommendations
Accreditation Phase
Activity 5
Perform Risk Assessment
Activity 6
Prepare Accreditation Recommendation
Activity 7
Make Accreditation Decision
Post-Accreditation Phase
Activity 8
Maintain Accreditation

C&A Process

The Accreditors Guide

presented by

Candice Stark, CSC
email cstark@csc.com

Why an Accreditors Guide?

- A common complaint of Certifiers as a group, was the lack of information for the Accreditor as to why s/he should be concerned with accrediting a system.

The Accreditors References

- Public Laws
- Executive Orders
- Office of Management and Budget (OMB)
- Federal Information Processing Standards (FIPS)
- The Rainbow Series
- Department of Defense, and Intelligence Community Directives

The Accreditors Guide

- Why, When, What, Who of Accreditation
- Knowledge Required by the Accreditor
- Authority, Accountability, and Responsibility of the Accreditor
- General discussion of threats, vulnerabilities and risks.
  - Residual Risks
  - Acceptable Risks

Why an Accreditors Guide?

- A common complaint of Certifiers as a group, was the lack of information for the Accreditor as to why s/he should be concerned with accrediting a system.

Accreditors Guide cont'd

- Miscellaneous Concerns
  - Acceptable level of effort
  - growth of an AIS into a Network
  - Sequential Development Accreditation
  - Granting of Waivers
  - Accreditation After a Successful Attack or Security Violation

Accreditors Guide cont'd

- Appendixes:
  - Security Laws, Executive Orders, and Directives
  - Common Threats
  - Assistance and Training Organization
  - Sample Accreditation Letters
  - Terminology, Acronyms and References.

Where do I find the C&A standards?

- **Executive Orders** http://library.whitehouse.gov/?request=ExecutiveOrder
- **Federal Information Processing Standards (FIPS)** http://www.nist.gov/itl/lab/fips/index.html
- **DISA Publications** http://www.disa.mil/pubs/pubs01.html
- **DISA Center for Standards** (TAFIM, EDI, DII) http://www.itsi.disa.mil
- **Rainbow Series** (subset) http://csrc.ncsl.nist.gov/secpubs/rainbow/
- **Listing of Standards by Standards Organization** http://bbs.itsi.disa.mil:5580/T1901

THE CERTIFICATION OF THE INTERIM KEY ESCROW SYSTEM

Ray Snouffer
National Institute of Standards and Technology
Building 820, Room 414
Gaithersburg, MD 20899
(301) 975-4436, ray.snouffer@nist.gov

The U.S. Government Key Escrow System (KES) provides for lawfully authorized access to the key required to decipher communications secured with products built in conformance with the Escrowed Encryption Standard, Federal Information Processing Standards Publication (FIPS) 185. This paper is intended for presentation at the 1996 National Information Systems Security Conference. The purpose of this paper is to describe the certification and accreditation of the Interim KES and provide an historical overview of the Key Escrow Certification Working Group's (KECWG) activities. The defined purpose of the certification working group is to perform a certification on both the interim and the final KES in accordance with the Guideline for Computer Security Certification and Accreditation (FIPS 102). FIPS 102 provides guidelines for computer security certification and accreditation of sensitive computer security applications. The National Institute of Standards and Technology (NIST) chairs the KECWG. In addition to NIST, the membership consists of the Department of Justice (DOJ), the Department of Treasury, the Federal Bureau of Investigation (FBI), the National Security Agency (NSA) and the Department of Commerce (DOC).

LESSONS LEARNED FROM APPLICATION OF THE DEPARTMENT OF DEFENSE INFORMATION TECHNOLOGY SECURITY CERTIFICATION AND ACCREDIATION PROCESS (DITSCAP)

Barry C. Stauffer
CORBETT Technologies, Inc.
228 N. Saint Asaph Street
Alexandria, VA 22314-2517
(703) 519-8639, staufferbc@aol.com

The DITSCAP establishes a standardized process, set of activities, general task descriptions, and a management structure to verify, validate, implement and maintain the security posture of the DII. The DITSCAP is designed to be adaptable to any type of Information Technology (IT) and any computing environment and mission. It can be adapted to include existing system certifications and evaluated products. It can use new security technology or programs, and adjust to the appropriate standards. The process may be aligned with any program acquisition strategy. Its activities can be integrated into the system life cycle to ensure the system meets the accreditation requirements during development and integration and continues to maintain the accredited security posture after fielding. While DITSCAP maps to any system life cycle process, its four phases are independent of the life cycle strategy. The DITSCAP's, four phases, Figure 1, are: Definition, Verification, Validation, and Post Accreditation.

- Phase I, **Definition**, defines the Certification and Accreditation Level of Effort, identifies the Designated Approving Authority, and documents the security requirements necessary for the certification and accreditation in a single document, the System Security Authorization Agreement (SSAA). Phase I focuses on understanding the mission, environment, and architecture to determine the security requirements and level of effort necessary to achieve accreditation.

- Phase II, **Verification**, verifies the evolving, or modified, system's compliance with the agreed upon security requirements.

- Phase III, **Validation**, validates the fully integrated system's compliance with the security requirements. Phase III concludes with full approval to operate the system, e.g., security accreditation.

- Phase IV, **Post Accreditation**, monitors system management, operation, and maintenance to preserve an acceptable level of residual risk. Phase IV includes those activities necessary for the continuing operation of the accredited system, e.g. change management, security management, and periodic compliance validation.

Phases I, II, and III are the DITSCAP process engine. The DITSCAP methodology permits the forward or backward movement between phases to keep pace with the system development or to resolve problems. Therefore the phases are repeated as often as necessary to produce an accredited system. The objective of these steps is to achieve agreement between the Program

Manager, DAA, and the Users Representative at each step of the process.

The DITSCAP was used as the basis for the certification and accreditation process in a recent government client server environment involving over 500 workstations. The application processes sensitive but unclassified information. This C&A effort was designed to meet the requirements of the new OMB A-130 Appendix III.

This presentation will discuss some of the lessons learned in the application of this new process. The discussion will include project planning, system analysis, requirements definition, requirements tracing, test planning, and testing.

Firewall Testing and Certification Panel

John Wack, NIST

This panel will examine a number of issues with regard to testing firewalls, including the following:

- what is the purpose of testing firewalls
- what sorts of tests
- how is the testing performed
- how can the results be interpreted

Firewalls are now being tested and rated by various organizations and journals. These ratings usually include some analysis of how "secure" the firewall is, i.e., how well the firewall lives up to its security claims and how well the firewall stands up to high traffic loads. But, some firewall experts disagree with the concept of rating firewalls for security, with one of the arguments being that the security of a firewall depends on many factors, some of which are difficult to test unless one performs testing on the firewall where it is installed. In other words, a firewall that may be deemed secure in a test environment may be quite the opposite in a different environment. At the same time, many find firewall testing and certification a useful metric for assessing firewalls and determining which firewall is best for their respective sites.

This panel will present several views of testing and certification, with representatives from industry and the DoD. The audience will be encouraged to participate with their own experiences on firewall testing and certification.

John P. Wack
National Institute of Standards and Technology
Computer Security Division, Bldg 820, Rm 426
820 West Diamond St, Gaithersburg, MD 20899
wack@nist.gov, 301-975-3359, fax 301-948-0279

The Trusted Product Evaluation Program: Direction for the Future

Moderator
Janine Pedersen
National Security Agency
JSPedersen@DOCKMASTER.NCSC.MIL

Panel Abstract:

This panel will include discussions about improvements and changes which are occurring in the Trusted Product Evaluation Program. Representatives from various initiatives within the Trusted Product Evaluation Program will discuss the overall strategy for the future of TPEP, including specific steps for moving the program to a new evaluation criteria, mechanisms for commercial advice to vendors, and new types of products which will be evaluated.

COMMON CRITERIA PROJECT IMPLEMENTATION STATUS PANEL

Panelists:

The panelists are representatives from the Common Criteria (CC) sponsoring organizations who are active participants in one or more of the current CC trial-use and implementation projects.

Lynne Ambuel
National Security Agency, US
ambuel@dockmaster.ncsc.mil

Murray Donaldson
Communications-Electronics Security Group, United Kingdom
mgdonal@itsec.gov.uk.

Robert Harland
Communications Security Establishment, Canada
rharland@cse.dnd.ca

Klaus Keus
BSI/GISA, Germany
keus@bsi.de

Frank Mulder
Netherlands National Communications Security Agency
mulder@nlncsa.minbuza.nl

Jonathan Smith
Gamma Secure Systems, United Kingdom
jsmith@gammassl.co.uk

Abstract

Common Criteria (CC) trial version 1.0 was completed in January 1996 and has entered into an active trial-use and implementation phase during 1996. Along with numerous trial evaluations of both IT security products and Protection Profiles against the CC by the sponsoring organizations in both North America and Europs, a number of related implementation projects have been initiated. These projects include:

- preparation of a common evaluation methodology,
- development of a framework for mutual recognition of the results of evaluation by the participating organizations, and
- study and development of prospective alternative approaches to evaluation.

In addition, extensive comments are being received from the IT security community review process. Expected output of all of this activity is a set of recommendations for revision of the CC to the definitive version 2 during 1997 and its acceptance as an ISO international standard.

The members of this panel represent the Common Criteria Implementation Board, the Common Evaluation Methodology Editorial Board, the Mutual Recognition Working Group, and the Assurance Approaches Working Group. The panelists will jointly discuss the CC trial version's structure and contents, the status and results to date of the trial-use and implementation activities, the planned future of the project, and the expected impact of all of this work on the US and international IT security communities.

Background

The Common Criteria Project is nearing the culmination of seven years of work in several nations to achieve a set of standard criteria for specifying IT security products and for performing evaluations on them. The goal is to provide a "level playing field" for both national and multi-national IT developers that will result in broader availability of IT products with known and trusted security characteristics for general use in both government and private organizations.

The original "Trusted Computer System Evaluation Criteria" (TCSEC) or "Orange Book", waas adopted by the US Department of Defense in 1985. This document has been used by the National Security Agency (NSA) for security product evaluation until the present. The known limitations of the TCSEC motivated NSA and the National Institute of Standards and Technology to embark on the Federal Criteria Project in early 1991 to create a more flexible set of criteria that can take into account advances in security technology and widespread inter-connectivity of computers. Federal Criteria draft version 1 was published in late 1992. Several European nations individually, then jointly, were working on their own criteria and evaluation programs during the same period, resulting in the initial publication of the Information Technology Security Evaluation Criteria (ITSEC) in mid-1990, with the current version delivered a year later. The Canadians also had begun their own criteria development activity in the late 1980's, and the last version of the Canadian Trusted Computer Product Evaluation Criteria (CTCPEC) was published in early 1993.

The US, Canada and the Europeans in 1993 agreed that it was time to work together to resolve any differences in approach and develop a single Common Criteria that could be contributed to ISO for use world-wide as an international standard. The Common Criteria Project thus began in the Fall of 1993.

Current Status

The CC was first published in rough draft for limited community review in mid 1994. It was extensively revised and again circulated, this time very widely, in late 1994. Based on input from the ISO committee also working on an international criteria, public review and comments, and the results of an international workshop, CC version 1.0 was published in January 1996. Part 1 of the CC had already been accepted by ISO as a working draft of its own criteria Part 1. Upon publication of CC version 1.0, ISO accepted all three principal parts of the CC as its second level "committee draft". This marked a major break-through, as for the first time there is a single internationally-accepted set of IT security criteria.

The CC Project Sponsoring Organizations created the CC Implementation Board (CCIB) to coordinate a variety of implementation activities, including trial evaluations, and prepare for the definitive version. The CCIB will collect and dispose of all identified problems and proposed changes to the CC during the trial-use period, whether from community review or from use of the CC for trial evaluations and preparation of evaluation methodology. The CCIB is also responsible for publicizing the CC and seeking its wide acceptance in the community of users, developers and academics. By the end of 1996, the CCIB will have collected all input on needed changes and will prepare a set of recommendations for preparation of the definitive version 2.0.

There are a large number of trial developments and evaluations underway based on CC version 1.0. In most of the participating nations, one or more trial evaluations of products are being conducted against the CC in parallel to their evaluations against the existing base criteria. In addition, CC-based Protection Profile requirement sets are being created for new products, such as firewalls and smartcards, as well as replacements for existing requirement sets in the TCSEC, ITSEC and CTCPEC.

The Common Evaluation Methodology Editorial Board (CEMEB) was also created in early 1996 to develop an agreed methodology that would represent the accepted way to perform product evaluations in the participating nations. There are three "legs" that support mutual recognition of each nation's security product evaluations:

- The CC itself, consisting of common requirements for security functions and assurance,
- A common approach or method for performing the evaluations, and
- Known competent evaluators to do the work in each participating nation.

Each nation or region performing product evaluations now has their own methodology. These methods have similar approaches and activities that constitute their evaluations, which must now be analyzed and the commonly-needed elements described. The CEMEB will develop and test these detailed evaluation methods.

A Mutual Recognition Working Group was formed.in mid-1996 to explore the legal, procedural and technical basis for each participating nation to recognize the IT security evaluation work of the other participants. This is a complex and potentially difficult topic because of differing legal structures, governmental policies, and current approaches. Currently, only a few bilateral agreements exist. It is expected that this group will continue to work over the next few years to put the broader agreements in place and resolve practical difficulties.

One CC-related group has been formed to move beyond the current evaluation-based assurance paradigm for commercially-oriented IT products. The major objective of the Assurance Approach Working Group (AAWG) is to investigate alternative approaches for gaining assurance that IT products and systems meet their security requirements. The group seeks to find, in the existing methods of product development or methods of validating them, alternate requirements to satisfy the assurances objectives expressed in the CC. This group is working to develop and test faster and more timely ways to provide trusted commercial products.

Plans

The ultimate plan for the CC is to gain information from application and study of the current version 1.0 to prepare a definitive version that can be turned over to ISO for use and maintenance as an international standard. Preparation of version 2.0 is expected to begin in early 1997. Continuing development of the common evaluation methodology and procedures for mutual recognition will proceed over the next few years, and the CC will be introduced into evaluation schemes in North America, Europe and perhaps elsewhere in the world.

Panel

Developmental Assurance and the Common Criteria

Moderator
Mary Schanken, *National Security Agency*

Panelists

Gene Troy, *NIST*
Klaus Keus, *GISA*
Yvon Klein, *SCSSI*
Stu Katzke, *NIST*

Abstract:

The traditional approach to obtaining assurance in the security features of information systems has involved extensive post-development evaluation of the completed product or system. The recent proliferation of commercial information systems has stressed our ability to evaluate products in a timely manner. At the same time, the globalization of commercial markets has motivated vendors to build to international standards such as the Common Criteria. These factors have encouraged the National Security Agency (NSA), the National Institute for Standards and Technology (NIST), and the North Atlantic Treaty Organization (NATO) allies to search for an alternate approach to determining whether information technology products satisfy security assurance requirements.

Current approaches to gaining assurance about information technology products meetings their functional requirements do not respond well to the market demands of developers and users. For commercial levels of assurance which appear to be acceptable by all users of commercially available products, an evaluation that is not completed within a reasonable time frame after product release is not useful to developers or users due to rapid and competitive changes in the information technology market place.

The Assurance Approaches Working Group (AAWG), composed of NATO, NIST, and NSA representatives, is developing a developmental-assurance framework that is mapped to the Common Criteria. The objective of this activity is to investigate alternative approach for gaining assurance that information technology products and systems meet their security requirements. It includes the definition of alternative assurance approaches to traditional evaluation and the building of alternative assurance packages. This activity seeks to find, in the existing methods of development or methods of validations, alternate requirements to satisfy the assurance objectives expressed in the Common Criteria.

The analysis will be performed initially for Evaluation Assurance Level 3 of the Common Criteria. An appraisal method will be defined, and candidates for the first appraisal against this Alternate Assurance method will be identified.

DARPA Research Panel 2:

Secure Networking and Assurance Technologies

Panel Chair: Teresa F. Lunt, DARPA

Panelists:
Karl Levitt, UC Davis
John McHugh, Portland State University
Steve Kent, BBN
Gary McGraw, Reliable Software Technologies
Doug Weber, Key Software
Lee Badger, TIS

Today's security solutions are being built for aging computing and communications technologies. Many of these solutions will not scale to the technologies of the future, and the future is just around the corner. For example, mechanisms that depend on cryptographic authentication of every packet in a data stream, that require frequent reference to distant directory servers to ascertain certificate validities, and that require lengthy appendages of signed certificates, may not be able to keep up with the speeds of new high-speed networking technologies or be at all appropriate for mutually authenticating software agents. Access control policies that were designed for a closed environment may not scale well to world-wide-web-style environments in which there are frequent interactions between unacquainted entities, or to a highly networked environment in which new alliances are quickly forged and terminated. The new phenomenon of cyberspace opens up privacy concerns that were not present in small, closed communities where one's every computing activity was not on display to the entire world.

In many cases, the focus on security must change from the individual end system to the network. For example, in intrusion detection, we must find analysis techniques that scale to very large systems (i.e., that do not require massive amounts of data to be collected) and that can produce reasonable results with partial data (since not all portions of a network are always visible). These systems should also be refocused to monitor network activity rather than exclusively end-system activity, and they should be made to work in a variety of networking technologies. We must better understand how to instrument our systems and networks so as to give us the requisite visibility. We also need to develop both intrusion detection and system management tools that can operate across administrative domains, or that may work with a network of other autonomous detection or management systems in a cooperative or hierarchical manner. These systems should be capable of dealing with extensive heterogeneity both with respect to the systems monitored and the detection and management systems themselves.

Most of the information infrastructure is going to be with us for a very long time. Telecommunications systems, electric power generation and distribution systems, financial systems, and transportation control systems will slowly evolve but will retain their legacy character through generations of technology improvements. In addition, many new critical systems, such as medical devices, defense command and control systems, and nuclear power plant control systems, are being constructed using commercial software products. We must begin work now to understand and deal with the risks of using commercial and legacy components in systems we depend on for our national well-being and personal safety.

We need strategies for working around the problems that are inevitably to be found in legacy and consumer-quality products. We need architectural "workarounds" to augment the strengths or compensate for the weaknesses of these components. DARPA is investigating whether security can be introduced into a system by developing security "wrappers" for certain system components. With this approach, wrappers would be used to introduce certain security functionality without altering the legacy code or the other system components that use it. The idea is to gain control over specific interfaces where a security function can be inserted. Such interfaces could be library calls, system calls, or other interfaces internal to a subsystem. For the approach to have any validity, it must be possible to ensure that all input to and output from the wrapped component can be intercepted by the wrapper; in effect, the wrapper becomes a reference monitor for the policy it enforces. This is the fundamental new assurance question for the approach.

This new approach requires new theories of secure composition of a system from components (including wrappers) and technologies for security integration. We must broaden the types of analysis that can be performed far beyond such narrow considerations as secure information flow for multilevel security. We must reason, for example, about how such diverse aspects of security as authentication, access control, and encryption contribute to overall system security when inserted into a system in various ways. In addition, our reasoning must allow for ignorance, empirical properties, or worst-case assumptions about legacy components. To support such reasoning, we must adequately specify the components; research is needed in order to understand what must be specified.

Security can be inserted in this manner to meet a variety of objectives. For example, it is easy to imagine how a wrapper could impose an access control policy on the wrapped component, or encrypt the outputs and decrypt the inputs of a components, or perform inter-component authentication, or perform message filtering. One could also design these wrappers to add security monitoring and intrusion detection capability. Ideally these wrappers should be designed so that the specific security solution is a modular part of the wrapper. This would allow the module to be replaced, for example, when it is desirable to use a stronger security solution. This should also allow multiple security modules, enforcing orthogonal policies, to be inserted in the same wrapper.

It has long been held by the security community that security must be designed into a system from its inception and cannot be added on later; we must investigate the feasibility of this new approach and discover how far and for what aspects of security it can be made practical.

The panelists explore these and other issues being investigated in the DARPA research program.

Secure Mobile Networks
John McHugh, Portland State University

Very little work has been done to integrate security and network-layer mobility into real systems that tackle the issues of secure enclaves. The work that we are undertaking will result in the development of a high performance Secure Mobile Network and insights into its use as part of the National Information Infrastructure.

Our goal is to produce a system that supports the establishment of secure enclaves or secure virtual networks among mobile workstations. We intend to combine a secure metwork layer including network layer authentication and encryption with robust Mobile-IP networking allowing secure mobility. Two-way tunnels will be used to allow remote networks or hosts to join a secure network across insecure topologies. We will investigate and design solutions for distributed access control protocols, and redundant systems needed for overcoming the single point of failure problems in the current Mobile-IP architecture.

In general, the IP community has limited experience with network layer security. Network layer security must be integrated with wireless Mobile-IP, another area in which the community has limited experience, and with other mechanisms needed to provide a suitably rich architectural environment that will deal with access control and other security issues as well as redundancy and other reliability issues. In attacking these problems, we will follow a rigorous engineering approach, guided by appropriate formal methods. We believe that protocols used in this sphere should be formally analyzed and their implementations subjected to rigorous software engineering techniques. Many network security problems are due either to faulty protocols or to flawed implementations or both and we hope to avoid these problems in our work.

Our initial system will combine a secure network layer, with Mobile-IP and two-way tunnels. A secure network layer has an operating system architecture component and a protocol component. For protocol components, we are following the IETF IPSEC working group recommendations as closely as possible in order to maximize the potential for technology transfer. Our protocol will provide authentication and encryption at the network layer.

The network architectural component includes access control and key management subsystems at the network layer. Outward and inward bound packet addresses will be looked up in the access and key management tables and appropriate actions, encryption, etc., will be taken. Access and key management daemons (application-layer processes) will allow for higher-level protocols and information exchange. We will design and implement a distributed access-control protocol. Such a protocol is analogous to current intra-domain routing protocols such as OSPF or RIP where clean separation of policy and mechanism exists between daemons and IP-level lookup tables.

Network layer security will be integrated with a Mobile-IP network architecture. The Mobile-IP architecture consists of a routing infrastructure containing three kinds of entities: Home Agents (HA), Foreign Agents (FA), and Mobile Nodes (MN). A single organization's MNs will typically belong to one or more IP subnets where the subnet address is topologically local to the organization. The HA is in charge of routng packets from the rest of the network to the MNs and tracks each MN via a registration protocol. When an MN moves from its home to a foreign subnet (or from one foreign subnet to another to another), it will send a registration packet to the HA via the current FA, which acts as a cell manager. After registration, the HA can forward incoming packets to the MN by encapsulating them in an outer IP wrapper with the FA as the destination. This is referred to as a "tunnel".

Currently, Mobile-IP assumes tunnels go one-way only from the HA to the FA. A recent CERT advisory has pointed out the dangers of local network addresses crossing from the outside to an inside network via a firewall. This appears to be a generic flaw in Mobile-IP and would prevent mobile systems from talking to local systems across current firewalls. We suggest that tunnels may be used as network bridges to allow remote mobile routers or hosts to convey their packets back across an insecure network to a secure router, thus forming a secure virtual network.

In addition to building an integrated secure mobile network that allows secure enclaves, we propose to investigate protocols that allow redundant Home Agents and Foreign Agents. Protocols that allow registration, handoff, and exchange of information between Home Agents are needed. A successful attack on a Home Agent or its failure for any reason could mean the catastrophic loss of a mobile network. A protocol for server redundancy should allow the mobile system to support more than one Home Agent.

Redundancy of FAs is also an important topic, since loss of a local FA might mean loss of communication with home or worse, complete loss of communication within a local cell. IP as currently construed assumes that the Address Resolution Protocol (ARP) cannot be used to establish communication between two hosts that are on the same link but are on different IP subnets (RFC 1122). Communication must be done through a router on the link (in Mobile-IP terms, the router would be the FA). We propose to develop an ad hoc protocol that would allow hosts within the same link to communicate directly where possible. Topologically, example systems could comprise a small mesh in which any system can address all other systems or a daisy chain in which each system can only address one or two other systems. It is always possible that systems might be able to talk to one system and not reach another; "can communicate with" is not transitive for radio.

Resolution of the routerless routing problem is a key factor in facilitating ad hoc networks. We want to be able to create these anywhere two or more MNs can communicate, whether or not a HA or FA is reachable.

We have established a Mobile-IP infrastructure in two buildings on the PSU campus. There are three agents (1 Home Agent (HA), 2 Foreign Agents (FA)) in our PCAT engineering building and one Foreign Agent in the Mill Street CS Lab building. Three graduate students, 4 professors (3 CS, 1 EE) and two staff members have mobile laptops. These run on a slightly modified version of the Free-BSD operating system.

In addition, we have established FAs at two off campus sites using modem connections via SLIP or PPP to connect to the campus network. In doing this, we have essentially managed to take PSU IP addreses to remote, disjoint locations. This allows Mobile-IP to be used to implement disjoint networks without requiring that internal routers actually know or support additional routes. It appears that this may permit a more efficient implementation of IP address space.

We have implemented a simple, but effective timestamp mechanism that counters most replay attacks while preventing replays from being used as a denial of service attack.

By the time of the conference, we hope to have made additional progress on several fronts. Our Mobile-IP implementation (MN, HA, and FA) will be available to interested parties by the first quarter of FY97. Check our web site for details (http://www.cs.pdx.edu/research/SMN/).

We are starting to integrate IPSEC with our Mobile-IP implementation, using Fortezza cards being supplied as GFE to rpovide encryption support. We will complement these with software encryption and possibly DES hardware encryption for the nodes for which we do not have Fortezza cards.

We will expend significant efforts toward making Mobile-IP more robust and secure through the provision of redundancy. There are three areas of work: 1. ad hoc routing, i.e., how MNs can route amongst themselves and also find paths to agents through other MNs; 2. redundant FAs; and 3. redundant HAs.

Adaptable Dependable Wrappers
Doug Weber, Key Software

The Adaptable Dependable Wrappers project is exploring a flexible way to build dependable distributed systems from software components. We are designing a prototype toolkit for generating adaptable dependable wrappers for the components of a system. We intend to test the flexibility of our approach by implementing the toolkit and using it to generate some sample distributed applications.

A *wrapper* for a software component forms a boundary layer between the component and all other components that interact with it. The purpose of the wrapper is to translate and filter the view these components have of each other's behavior.

A *dependable wrapper* imparts critical properties to each component that it wraps. For our purposes, "dependability" includes both survivability and security. Some dependable wrappers have been built before, but without the flexibility of our approach. A survivable wrapper typically wraps a group of replicas of the component, coordinating the replicas for fault tolerance. Security wrappers have been used for many purposes, including authentication and access control.

We are generalizing this previous work by creating dependable wrappers that are also *adaptable.* We mean "adaptable" in a general sense, including both *configuration* at compile time and *reconfiguration* at runtime. An engineer will configure a dependable component wrapper framework at compile time by choosing from a library:

- algorithms and protocols that support critical properties he specifies;
- a design that will work efficiently in the component's environment.

At runtime the wrapper will reconfigure itself automatically when it interacts with other components. An adaptable dependable wrapper:

- can learn the specification of another component;
- can decide whether the other component's specified critical properties are sufficient to support its own;
- can decide whether to trust that the other component actually implements its specification;

- can learn from the other component new protocols that must be used to guarantee critical properties;
- offers information about its own properties to other components.

Adaptable dependable wrappers offer the following advantages over existing technology:

- The wrappers can be used to gain security and survivability in a wide variety of distributed systems. Components can be wrapped specifically to support each system's requirements.
- A component of a long=running system can be replaced (for modification, upgrade, or with a new application) without restarting the system. Replacement is easier and arguably safer than in current distributed systems because a new component teaches others about itself.
- A survivable system can degrade gracefully after massive failures by weakening its dependability specifications. The surviving components may be able to continue functioning by learning to interact with new, less dependable, components chosen from a larger pool.

The Adaptable Dependable Wrappers project is part of DARPA's Information Survivability program.

Generic Software Wrappers for Security and Reliability
Lee Badger, TIS

Very large-scale information systems are increasingly built by combining numerous independently developed software components. Components may be programs, linkable code libraries, and, increasingly, network applets based on emerging software frameworks (e.g., CORBA, OLE, CGI, Tcl, Java). While use of independent, and standardized, components reduces cost, component failures and unintended interactions among components seriously threaten the reliability and security of information systems that use them. Components are often engineered for "commercial" assurance but then are deployed within critical systems requiring high assurance. Of particular concern are network applets that bring new power to rapidly deploy information systems but also add risk: applets often exchange interpreted data, which makes them highly vulnerable to corrupted data. Applets may also be dynamically reinstalled: this potentially exposes information systems to flaws in future as well as current software components.

Dramatic advances in information system security and reliability will require techniques both for limiting the damage that can be caused by individual components and also for adding reliability features tailored to system mission requirements. A variety of techniques (e.g., Internet firewalls, extensible operating systems, fault isolation) control or enhance component interactions, but these techniques are too costly, not generic, or provide inadequate support for coordinating security and reliability policy data.

This project will develop techniques and tools for specifying and implementing generic software component wrappers. Generic software wrappers will intercept component interactions and bind them with additional functions that implement practical security (e.g., restricting, filtering) and reliability (e.g., redundancy, crash data recovery) policies. We believe that a successful wrapping technology must: 1) wrap existing components, 2) accommodate a large number of software interfaces and policies, 3) work in numerous execution environments, 4) be optional and consistent with high performance, and 5) be capable of high assurance.

This project will develop a prototype Wrapper Development Framework to demonstrate practical software-wrapping technology that meets these criteria. The wrapper development framework will include a Wrapper Definition Language (WDL), a Generic Wrapper ToolKit, a Wrapper Support Interface, and two systems that implement it: a wrapper-supporting UNIX prototype and a wrapper-supporting Java prototype. The Generic Wrapper ToolKit will implement wrappers expressed in WDL and will provide tools to wrap and unwrap selected components at runtime. The Wrapper Support Interface will define a modest level of generic wrapper support (necessary for high assurance) suitable for standardization and inclusion in mainstream execution environments.

This project will implement wrapper support in both a kernelized UNIX and an interpreted Java environment to build confidence that the approach is general and that WDL wrappers are portable. By demonstrating practical, generic software-wrapping technology, this project seeks to provide a basis for significant security and reliability increases in large-scale information systems based on reusable software components.

Defining an Adaptive Software Security Metric from a Dynamic Software Fault-Tolerance Measure

Gary McGraw, Anup Ghosh, & Jeff Voas
Reliable Software Technologies Corporation
21515 Ridgetop Circle, Suite 250
Sterling, VA 20166
{gem,anup,jmvoas}@rstcorp.com   http://www.rstcorp.com

Abstract

The original computer security defense strategy, circa 1970, was appropriately termed "penetrate and patch." At that time, defense was entirely reactive — something that happened only after an attack was detected and some damage had already been inflicted. Penetrate and patch was followed by a series of more advanced defensive techniques (*e.g.*, real-time intrusion detection tools, COPS, and SATAN). Unfortunately, a recent proliferation of sophisticated threats has caused defensive security schemes to come full circle, back to where they began twenty-some years ago. Penetrate and patch has once again become the status quo.

This abstract briefly describes work-in-progress under ARPA contract number F30602-95-C-0282, "Quantifying Minimum-time-to-intrusion Based on Dynamic Software Safety Assessment". We have developed a software metric that is currently being implemented to quantitatively assess information-system security and survivability. Our approach — called Adaptive Vulnerability Analysis (AVA) — exercises a piece of software (in source-code form) by simulating both malicious and non-malicious attacks that fall under various threat classes. AVA can be used to determine whether such threats undermine the security of the system. This approach stands in contrast to common security assurance methods that rely on black-box techniques for testing completely-installed software systems. AVA does not provide an *absolute* metric (such as mean-time-to-failure). However, it can be used as a relative metric, allowing a user to compare the security of different versions of a system, or to compare non-related systems with similar functionality.

AVA derives from models that were developed for assessing software fault-tolerance — in particular, a model used for Extended Propagation Analysis (EPA). Implemented models of EPA are automatic systems that use fault-injection methods to predict how software systems will behave when faced with anomalous circumstances such as: (1) simple and complex programmer errors, (2) rare but correct input data, (3) corrupted input data, and (4) failed hardware signals. In this ARPA-sponsored project, we are extending and adapting the functionality of EPA software-analysis models so that we will be able to predict the impact of an additional important class of anomalous circumstance on software systems — namely, malicious threats.

References

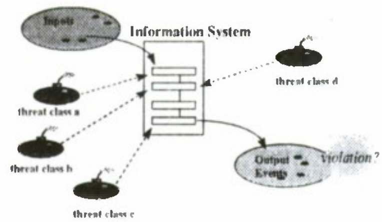
Voas, Jeff, Gary McGraw, & Anup Ghosh. Defining an Adaptive Software Security Metric from a Dynamic Software Failure Tolerance Measure. Reliable Software Technologies Technical Report. March 28, 1996. Sterling, VA.

Voas, Jeff, Anup Ghosh, Gary McGraw, Frank Charron & Kieth Miller. (1996) Defining an adaptive software security metric from a dynamic software failure tolerance measure. In the *Proceedings of the Ninth Annual Conference on Computer Assurance*, pages 250-263. June 1996.

Defining an Adaptive Software Security Metric from a Dynamic Software Fault-Tolerance Measure

Gary McGraw
Reliable Software Technologies
gem@rstcorp.com
http://www.rstcorp.com

The Big Picture

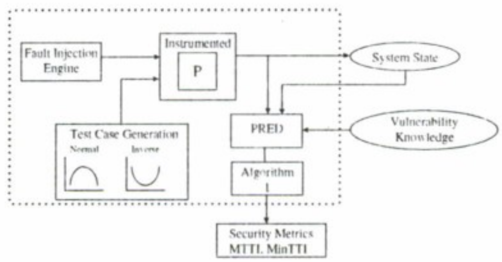


Starting Point: A Fault-injection based Fault-tolerance Model

Extended Propagation Analysis (EPA) aids safety assessment, in several ways:

Predicts the likelihood that software faults and hardware failures can propagate to *unsafe* or *undesirable* outputs of the software.

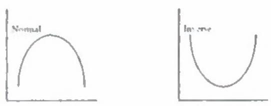
Prototype Tool Design



Test Case Generation

Automated test case generation supports:

- normal operational profile
- unexpected or "garbage" input
- inverse operational profile



Predicate Specification

Intrusions are defined via logical predicates

- unauthorized read access
- unauthorized write access
- denial of service
- others defined by user according to application

Fault-Injection: A Means of Simulating Errors and Threats

• Assesses the robustness of a system and how well it recovers.

• Used in physical world for years.

• "What-if"?

• The more "what-if" games you play, the more confident you are that your system can overcome anomalous situations.

X-time-to-Intrusion

- A probability estimate ψ_{aIPD} (from the program executions) for how often an intrusion occurred
- Given a number of program executions per unit time, you can derive a *mean* or *minimum time to security violation.*

Technical Summary

- Fault-injection methods have worked well for years in the *physical* world.
- Safety and Security are unique but similar in certain respects --- we want both!
- Unique fault-tolerance assessment model.
- Quality of prototype will be dependent on those prior threats that are simulated by the ARPA innovation.

http://www.rstcorp.com

MoIPS & IRIS
Internet Security R&D

Dr. Stephen Kent
Chief Scientist- Information Security
BBN

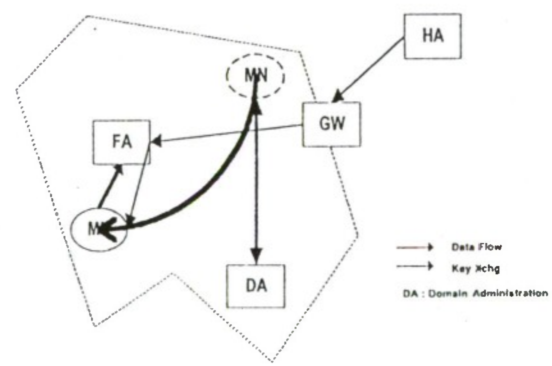
Outline

- Global Mobile IP Security (MoIPS)
- Internet Routing Infrastructure Security (IRIS)
- Summary

MoIPS Project Overview

- Goals
  - support security associations among all mobile nodes (MNs), home agents (HAs), foreign agents (FAs), and correspondent hosts (CHs)
  - develop mobile IP extensions to support domain-based host admission control & fast (1 second) handoffs
- Approach
  - integrate mobile IP with IPSEC (AH & ESP) protocols
  - develop DNS-based public key infrastructure for managing certificates of internet nodes
  - develop domain-based mobile IP for fast registration & localized location updates

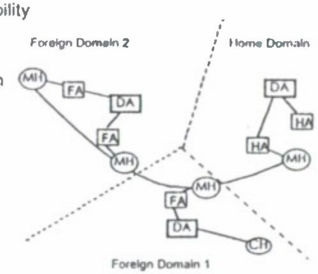
MoIPS Basic Operation



MoIPS Concept

5

- Key concepts
  - internet domain hierarchy for host mobility hiding & admission control
  - secure IETF mobile IP for inter-domain mobile node registration
  - fast intra-domain registration for intra-domain mobile node location updates

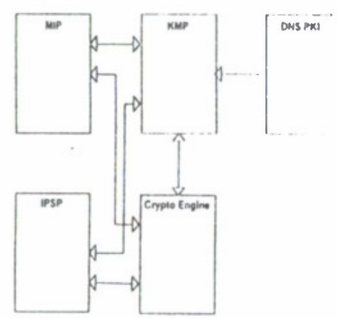


DA = Domain Administration

MoIPS System Modules

6

- IETF-compatible Mobile IP
- HMAC (MD5) authentication
- Unidirectional, low overhead key management
- X.509 v3 certificates, v2 CRLs
- DNS certificate/CRL records
- Tunnel mode AH + ESP



The IRIS Program

7

- Focus on advanced routing protocols
  - Inter-Domain Routing Protocol (IDRP)
  - Nimrod
- Security concern: countering denial of service attacks
- Perform security requirements analysis
- Develop countermeasures based on integration of cryptography & protocols
- Demonstrate effectiveness & practicality of the countermeasures

Security Requirements Analysis

8

- Security requirements represent "correct operation"
- Attack models
  - attacker capabilities
  - attack leverage
- Countermeasure capabilities
  - performance costs
  - implementation & administration costs
- Hybrid approach to requirements
  - top-down
  - bottom-up

9

Nimrod Security Requirements

protocol \ security service	Data Origin Authentication	Peer Entity Authentication	Rule-Based Access Control	Identity-Based Access Control	Connectionless Integrity	Sequence Integrity	Confidentiality	Non-Repudiation
Neighbor Discovery	X				X	X		
Agent Discovery	X				X	X		
Locator Procedures	X			X	X	X	X*	
Map Procedures	X				X	X	X	X
Adjacency Procedures	X			X	X	X	X*	
Route Requests	X				X	X	X*	X
Data Forwarding	X				X	X		
Path Setup/Accept	X		X	X	X	X	X*	X
Path Teardown	X		X	X	X	X	X*	X
Path Status	X				X	X		
Path Ack	X				X	X		

10

Selected Countermeasures

- Digital signatures
- X.509 certificates and CRLs
- Keyed hash functions
- Timestamps
- Shared secret protocol
- IPSEC AH with anti-replay

11

Testing & Demonstration

- Instrument routing implementations to measure performance and to add status displays
- Measure performance before countermeasures
  - under simulated, non-hostile scenarios
  - under selected attack scenarios
- Measure performance after countermeasures
  - under simulated, non-hostile scenarios
  - under selected attack scenarios
- Demonstrate countermeasure effectiveness using simulated attack tools

12

Summary

- Both programs have similar, top-level strategy for addressing (new) Internet security problems
- Different primary security service foci
  - MOIPS: authentication & sequence integrity
  - IRIS: communication availability
- Some overlap in countermeasure technology
  - X.509 certificates & CRLs
  - keyed hash functions
  - timestamps
  - IPSEC

Generic Software Wrappers for Security and Reliability

Lee Badger
(badger@tis.com)

Trusted Information Systems, Inc.
3060 Washington Road (Rt 97)
Glenwood, MD 21738
(301) 854-6889

Problem Statement

- Critical systems increasingly use generic software components:

  Global Command and Control (Sun Solaris, Oracle DB, X windows)
  Military Health Services (MVS, MS-DOS, UNIX, VMS)
  Government Intelligence Workstation (X windows, WWW)
  Joint Task Force Operations (concept to capability in weeks)
  . . .

- but generic components may not be secure or fault tolerant,
- and an error in one component may cascade to others, possibly subverting an entire system.

Solution Strategy: Better Composition Techniques

- Traditional composition techniques:

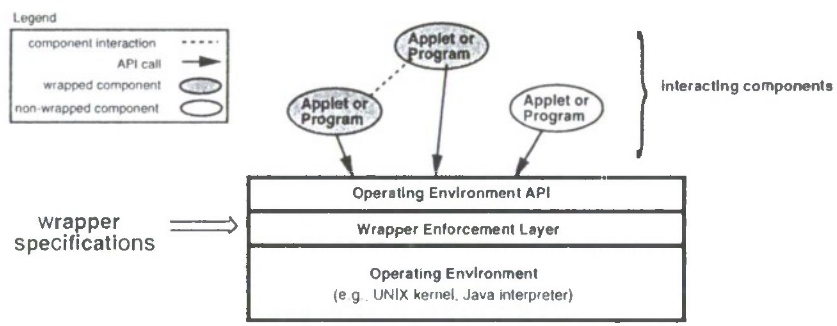
  Operating Environment APIs | Dynamic Link Libraries | Object Frameworks
  Network Protocols | Applet and class loading

  leave systems vulnerable.

- A Generic Software Wrapper is a policy-driven component mediator that selectively intercepts component interactions to:

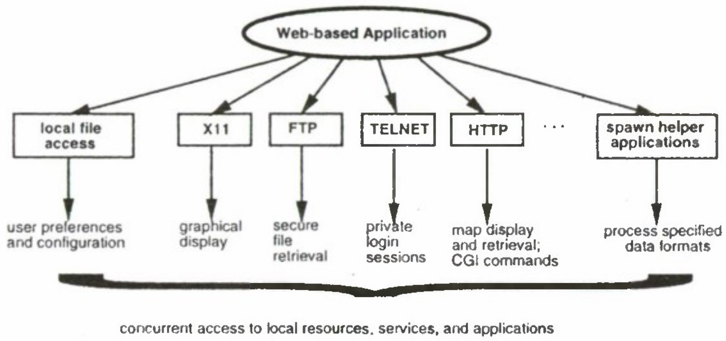
  control propagation of errors
  add security/reliability policies

Wrapper Architecture



- the wrapper enforcement layer processes wrapper specifications and keeps track of which components are wrapped
- wrappers mediate and selectively enrich component interactions

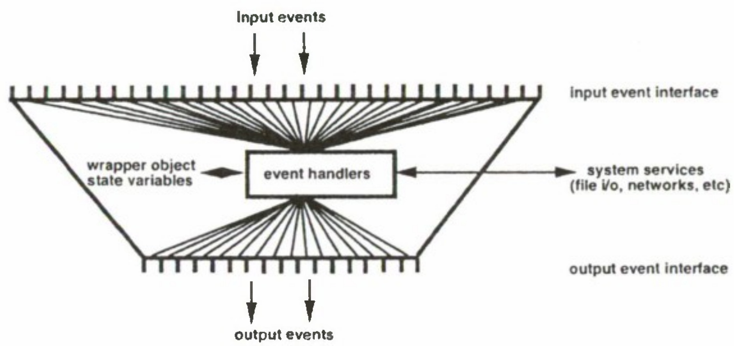
Wrapper Requirements



Control/augment applications that:

- are richly connected and web-based
- employ multiple network protocols, services, host resources
- are characterized by numerous transient relationships

Wrapper Definition Language



- API function intercept
- Event abstraction
- Attribute management
- Policy models and composition of properties

Project Plan

- Wrapper Definition Language (WDL)
- Generic Wrapper Toolkit
  - prototype WDL compiler
  - wrapper management tools
- Wrapper Support Interface
- Prototype implementations:
  - BSD/OS UNIX
  - Java Runtime

Using Security To Meet Business Needs: An Integrated View From The United Kingdom

Panel Members:
Chairman: Mr. Alex McIntosh, Chairman, PC Security Ltd.

Members:
Dr. David Brewer, Chairman, Gamma Secure Systems Ltd.
Mr. Nigel Hickson, Department of Trade and Industry
Mr. Denis Anderton, Barclays Bank PLC
Dr. James Hodsdon, CESG
Mr. Michael Stubbings, GCHQ

Theme:

The use of risk management techniques in the identification, accreditation, and maintenance of appropriate security profiles for single organization systems dispersed across a wide range of sites. Examples to be drawn from the defence, intelligence, governmental, financial and commercial sectors, together with the relevance for Generally Accepted System Security Principles, and their relationship with national UK policy.

Panel Statement:

The majority of information protection issues faced by most companies and government agencies of whatever size are the result of the increasing use of Information Technology. Technology creates the problems. Technical solutions exist to fix the problems, but technology itself isn't enough. The company or agency must have a security policy and security strategies which are all well-thought out and documented. It must ensure the implementation of its policy by executive management support and well-managed programmes. Such a management programme must be fully integrated into the overall business objective of an organization. Managers often have to make trade-offs between different business objectives, and information security issues are not immune from such considerations. Like all other business activities, information security must make its contribution to the well-being of the organization as a whole.

This session brings together a group of UK practitioners to discuss the management issues and requirements, and how they are being addressed by UK government and industry.

PANELLIST'S STATEMENT: MR A M McINTOSH

Mr A McIntosh,
PC Security Ltd.,
Windsor House,
Spittal Street,
MARLOW
Buckinghamshire
SL7 3HJ
United Kingdom

Tel: +44-1628-890390

Alex McIntosh is Managing Director of PC Security Limited (PCS), a specialist computer security company offering access control, encryption and management solutions for information protection. The company is headquartered in Marlow, UK, and has recently opened offices in the USA. The Stoplock range of products for PCS and LANs is unique in being certified to ITSEC Level E3.

PCS has business partnerships with EDS, Harris Computer, ICL and Motorola, for the integration and marketing of its products.

McIntosh has been in the computer industry for 35 years, previously with IBM, where latterly he was a senior executive in IBM Europe. He is Chairman of the ITSEC Scheme Industry Working Group, and sits on a number of government sponsored committees.

PANELLIST'S STATEMENT:
DAVID F C BREWER, Gamma Secure Systems Limited

Dr. David Brewer
Gamma Secure Systems Limited[1]
Diamond House
149, Frimley Road
Camberley, Surrey
GU15 2PS
United Kingdom

Tel: +44-1276-691415 Fax: +44-1276-692903
E-mail: dbrewer@gammassl.co.uk

Quite apart from the Cabinet Office Review of Protective Security (RPS), there have been other changes which have propelled Information Security (IS) to the top of UK MoD's agenda as a business risk management tool. This parallels recent changes in the commercial arena where the marketplace is demanding greater assurance of secure operation from payment and information services where high value is at stake (see Figure 1). The question that I would like to raise is whether this heralds a convergence between commercial and defence IS approaches or whether fundamental differences still remain.

Prior to RPS, the over-arching policy of 'risk avoidance' compelled the MoD and other government departments to seek multi-level security solutions that where undoubtedly beyond the state-of-the-art at the time. Understandably this led to some spectacular failures in the late 1980s and prompted a complete IS re-think. Moreover, the advent of the home computer in the early 1990s meant that MoD staff enjoyed computer power at home that was vastly superior to that which they could ever enjoy from a bespoke solution in the office. Clearly, the traditional acquisition methods for military hardware such as tanks and ships were rapidly becoming inappropriate for software intensive projects. In the UK, we concluded that for many applications, it might be possible to utilise commercial-of-the-shelf (COTS) technology. There were two imponderables: what level of security could we get? and could the products be reliably integrated together? Accordingly, in 1992 the MoD

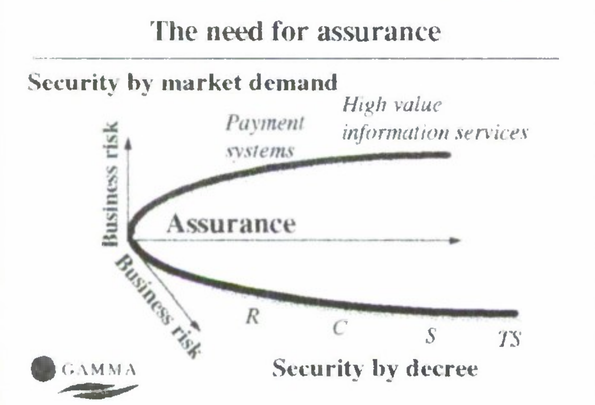


Figure 1

[1] Gamma is a leading UK information security consultancy which creates products and services to help customers gain assurance that their information security needs are met.

commissioned the Secure Open System Technical Demonstrator Programme (SOS TDP) to find out.

The adoption of RPS in April 1994 heralded a move from 'risk avoidance' to a 'risk management' approach to security. The threat had changed with the demise of the Soviet Bloc, but so had the theatre of operations. The concept of tension, transition to war, and then war itself, with no apparent transition back to peace, had been replaced by a cycle of rapid deployment, engagement, and re-deployment, followed by a comforting return to barracks. The need to rapidly acquire and assemble new information technology (IT), to meet the demands of some new deployment such as the Gulf War, were more important than ever before. Early SOS TDP results, coupled with the pragmatic approach to IS afforded by the RPS, indicated that this might be simply achieved with COTS products with 'Orange Book' class C2 functionality. However, other developments were to change that view entirely.

Firstly, a series of studies, that considered the IS requirements for a comprehensive range of MoD systems, concluded:

1. Most MoD systems operate in a system-high mode (i.e. user clearances exceed data classification), but labelling is required.
2. There are at least three different variations of the 'system-high with labels' policy, each one characteristic of a different type of MoD business.

Secondly, there was a major organisational change within the MoD which brought together the acquisition of peacetime and operational systems. This brought home the realisation that the peacetime and operational tasks were often closely related, and in some cases performed by the same people. The threats are different in the two environments but the information and ownership is the same. This highlighted the fact that the IS requirement was dependent on the business requirement. Indeed, this business requirement has been recently reviewed, taking a top down approach to determine how all current and future systems/functional areas should work together to form an effective whole.

Scheme Take-up (by vendor)

1990	1992	1993	1994	1995	1996/97
Amdahl	*Bull*	*CA*	*Oracle*	*IBM*	*Argus*
ICL	*Hitachi*			*Informix*	*Concurrent*
	Siemens				*DG*
					Digital
					EDS
					Harris
					HP
					Microsoft
					Novell
					Sequent
					Sun
					SCO

Source: UKSP06, October 1995

GAMMA

Figure 2: Date (or anticipated date) of ITSEC certification for various products

Finally, there have been major advances in technology and changes in MoD's use of IT. In particular, interest in CALS and the Internet has highlighted the need for firewalls and cryptographic-based controls, such as electronic signatures and non-repudiation services. Moreover, 1995 saw a tremendous uptake in the UK ITSEC evaluation and certification scheme, as indicated in Figure 2. Soon, I would predict that trusted CMW-like platforms will become commonplace in the home environment to safely launch Visa/Mastercard and electronic cash payments, rather like a CMW can be used in a bank to enforce the traditional 'check and release' function. In some sense, therefore, secure commercial IT has overtaken the SOS TDP, which has a become a conduit for its introduction into MoD systems.

In view of these developments, MoD is in the process of rationalising its approach to IS in two ways: by adopting a common risk management approach across all functional areas, and by preparing properly for the Information Age. It is in this latter respect that IS as a business risk management tool will really come into its own. The MoD is currently developing a new approach to specifying IS requirements as a characteristic of 'business domains'. These domains transcend the traditional IT boundaries to take account of user awareness, and physical and procedural measures. As such, the approach lends itself to business risk management and should be extendible to embracing concepts such as British Standard BS7799, a forerunner of the Generally accepted System Security Principles (GSSP) initiative, as well as ITSEC, the Common Criteria and GSSP itself. Of perhaps greater interest is that these domains may interface with 'commercial domains', such as payment systems, and information systems such as CNN.

PANELLIST'S STATEMENT: NIGEL J HICKSON, DTI

Mr N J Hickson,
Department of Trade and Industry,
151 Buckingham Palace Road,
London
SW1 9SS
United Kingdom

Tel: +44-171-215-1315
Fax: +44-171-931-7194

The Department of Trade and Industry in the UK has a fairly simple, if not somewhat ambitious mission. Basically it is to do whatever is necessary to ensure that UK business remains competitive in relative terms and is able to take its proper place in the global economy. In doing so the ability of industry to respond to technological innovations is all important: businesses that do not, or are not able to adapt to new trading conditions (whether procedural or technical) will rapidly lose their competitive edge. Within such an innovation process, the evolutionary spread of information technology has been a significant factor for nearly all businesses. The "spread" of IT from behind the locked doors of the computer room with "Keep Out" written in red, to every employee's desk (and to the laptops which many of us lug around) has revolutionised the way companies do business both internally and with their customers and suppliers. It has given many firms significant new market opportunities and has enabled small firms (to whom IT only ten years ago would have been an unnecessary luxury), in particular, to compete on an equal basis with their larger cousins.

In maintaining (and indeed, promoting) the effective and efficient use of IT as a significant competitive driver, it is only natural that the DTI should be concerned then with any factors that limit or militate against such efficient and effective use. And therefore it makes sense for us to be concerned with the security of information and IT systems. For the unavailability of these systems (or the data they handle) and compromises to either the integrity or confidentiality of such data, will inevitably lead to a degradation of the service offered by the IT.

DTI is, therefore, extremely concerned that all businesses, no matter how large or small, are equipped with the necessary tools (and these include guidance and advice) to be able to deal with information security issues. Given the importance of IT to organizations, and therefore the criticality of keeping it running, it is only sensible that information security is dealt with as a business and a management issue rather than as simply an *irritant* that can be addressed by technical solutions on their own. To achieve such a focus is not, however, trivial. For too long information security (or rather computer security as it is often mistakenly referred to) has been presented by both Governments and security professionals as a technical issue. The former has, up till now, been rather too concerned about confidentiality (at the expense of integrity and confidentiality) while the latter have been rather too keen to advocate expensive technical solutions.

It is because of the perceived need to "shift" this balance that DTI have, for some time, been co-operating with industry to produce guidance and advice aimed at business managers within organizations. Following on from the introduction of BS7799 (The Code of Practice for Information Security Management) we have introduced a number of guides to try and "grab" the attention of our target audience. These have included the Computer Assurance Guidelines (an attempt at presenting a risk based approach to information security) and the Internet User's Guide to Security, aimed at those companies about to embrace the Internet for the first time. We are also launching this autumn (just ready for Baltimore) two guides on the “Classification of Information” which enables organisations to classify, and then protect, their information based on its importance, and sensitivity, to the organisation.

In addition to promotion and guidance DTI also organises business briefings and executive lunches to try and secure the attention of the busy executive. We have, for example, just formed an Information Security Round Table bringing together senior executives from both the private and public sectors. We are also, not surprisingly, talking about the “encryption issue” with senior business representatives.

In my short address at Baltimore I will attempt to convince you of the importance of treating security as a business issue, and will give a short update on recent DTI initiatives, give out a few free booklets (including the new ones on “Classification”) and finally engage in vigorous debate with my panel colleagues and, of course, with you, the audience.

PANELLIST'S STATEMENT - DENIS ANDERTON, BARCLAYS BANK PLC

Denis Anderton
Barclays Bank PLC
Head of Risk Management
Payments and Cash Management Services
Floor 4C
Barclays House
1 Wimborne Road
Poole, Dorset, BH15 2BB
England
Tel: 44 1202 403363

My role within Barclays Bank is that of Head of Risk Management for a business unit known as Payments and Cash Management Services. This business unit is responsible for all of the payments and cash management products that the Bank provides to its' corporate and retail customers. These include cash, paper cheques and credits, domestic and international collections and payments as well as the Bank's electronic banking products.

To give an idea of the scale of our operations, our main centre processes in the region of £60bn of high value payments on peak days.

As one would expect, all aspects of risk management are very relevant to us. In particular the requirements for extremely high levels of availability and integrity of information are becoming more significant as our industry moves rapidly into the electronic delivery of products. The development of real time gross settlement for payments systems also requires us to strive for 100% availability of our networks and supporting applications. We also operate in a highly competitive sector of industry which necessitates cost effective solutions.

We believe that risk management must be driven from the top down by the people charged with running our business i.e. our Managing Director and his executive management team (which includes myself). Risk management objectives and activities must be based on business objectives and requirements, and must be led by business management. In fact risk management should be a key driver in determining business strategy.

To ensure this is achieved we have developed a strategy for risk management, which covers business and legal risks, as well as IT. This includes such things as a formal risk management structure and a highly structured approach to staff awareness.

Fundamentally, I believe that good risk management is a cultural issue - it's an attitude of mind. One of our themes for risk management is that "Awareness is the most effective countermeasure". If our people understand the risks and the effects of poor performance, they will typically develop the appropriate controls in their every day work. Given the right leadership this can and is being achieved in Payments and Cash Management Services.

I look forward to discussing these issues with the delegates.

PANELLIST'S STATEMENT: DR JAMES HODSDON, CESG

Dr James Hodsdon,
Room 2/0506,
Communications-Electronics Security Group,
Fiddler's Green Lane,
CHELTENHAM
Gloucestershire
GL52 5AJ
United Kingdom

Tel: +44-1242-221491 ext 4195

The Communications-Electronics Security Group (CESG) is the UK government's lead authority on Infosec technical issues; administratively it is part of the Government Communications Headquarters (GCHQ). During the UK's 1994 Review of Protective Security, which looked at security issues throughout the government and armed forces sectors, CESG took an active part in meshing best practice in the Infosec areas, which already included several examples of "graduated response", with the new government-wide guidelines on risk management.

Since 1994, CESG has been a major contributor to the follow-through work generated by this Review. This work has been one of the chief tasks of the policy team which I head up, and has led to a complete overhaul of the UK government's top-level policies and guidance on IT and communications security. The outcome has been new documents which aim to describe the risks, the issues and the solutions in terms which **any** government user can relate to. This is important because our customer community is no longer restricted to the classic "Classified" users. The new classification ("protective marking") system in the UK intentionally embraces **all** official assets needing protection. We have to generate guidance that reflects all the different risks not just in military but also in normal public service environments. This harmonisation process has been a highly beneficial sanity check; too many of the rules devised for good reason in the old classified era had become fossilised and were only marginally relevant to today's government office environment and technologies.

There is little legislative framework or enforcement apparatus surrounding Infosec practice within the official sector in the UK. Generally speaking, new protective security policy cannot simply be imposed from the centre. Any proposed new Infosec policy has to be explained and demonstrated to be a credible realistic method for managing the risks. It also has to be endorsed by a committee system in which the security authorities and the user communities (military and civil) all have a voice. This consensus system, with all sides "buying in" at the start, is what gives the policies their practical strength.

In the risk management era, one of CESG's primary tasks is to explain to users what the Infosec risks are, and what the choices are for managing those risks. This is a far cry from the old days of "These are the rules and this is the kit which the rules dictate". It is also the only way to go when technology and risks are in constant change.

PANELLIST'S STATEMENT: MICHAEL E J STUBBINGS, GCHQ

Michael E J Stubbings
Room A/1411,
Government Communications Headquarters,
Priors Road,
CHELTENHAM
Gloucestershire
GL53 7PN
United Kingdom

Tel: +44-1242-221491 ext. 3273

The Government Communications Headquarters (GCHQ) is an autonomous department within the United Kingdom Civil Service, and is one of the UK's three intelligence agencies. Like all large organizations it has to give careful attention to the cost-effectiveness of all aspects of its operations. Its approach to the handling of these pressures in the field of computer security is documented in the paper on accreditation which accompanies the panel proceedings. Its security interest and position as one of the UK's most technically-orientated departments has offered many opportunities to consider the implications of different approaches to computer security.

It is normal practice within GCHQ to separate the issue of confidentiality from those of integrity and availability. The latter two are dealt with by individual projects and are not subject to review by GCHQ security staff except insofar as the security profiles of any measures are concerned. In GCHQ, effectiveness in achieving integrity and availability is the business of the project staff, not of the security department. My area is strictly that of confidentiality, and my role is as senior computer and communications security accreditor, leading a team of 5 people. All of us are full-time IT security consultants covering all aspects of GCHQ's work.

GCHQ has been foremost in the UK in adopting and implementing the risk management approach mandated by the Prime Minister for UK government use. This approach is described in the Review of Protective Security, published by the Cabinet Office. The adoption of management disciplines from the commercial world has given further impetus to the consideration of cost-effectiveness at all stages of a system's development and use.

We are well aware that the UK is not alone in considering these issues: last year's presentation of the proposed revision to the Office of Management and Budget's Circular No. A-130 Appendix III demonstrated some of the same principles, particularly in its definition of 'adequate security'. I, along with the other panellists, look forward to discussing these issues with delegates.

NISS PANEL
Security APIs: CAPIs and Beyond

ABSTRACT: Last year NSA issued a set of recommendations for Cryptographic Application Program Interfaces (CAPIs). Since that time, updates have occurred to these CAPIs and Microsoft has adopted their own CAPI. In addition, implementation efforts are underway to validated these recommendations. Now that CAPIs are making cryptography accessible, the need for similar access to security services is growing. These services include certificate management, authentication, and key management. All of which are crucial to the current Public Key Infrastructure (PKI) activities.

Today's panel will include representatives throughout the community that are dealing with issues regarding Security APIs (including CAPIs) and PKI.

Panelists	Company	Topic
Amy Reiss (Chair)	NSA	SSAPI Overview and Strategy
John Centafont	NSA	FORTEZZA and CAPIs
TBA	Microsoft	Microsoft Internet Security Framework
Lawrence Dobranski	CSE	PKI
David Balenson	TIS	ICE Update

SSAPI Overview and Strategy: An Application Program Interface (API) is an interface that enables application developers to call and utilize specialized functions within their applications without having to be experts to those specialized functions. Using this approach, security APIs can be developed in order to provide application developers the capability to easily incorporate security into their application without having to be security experts. In addition, APIs provide the capability to plug-and-play the underlying security mechanisms and cryptographic tokens. There are three areas of particular importance, Cryptography (CAPI), Certificate Management (CMAPI), and Key Management (KMAPI).

CAPI: The original NSA CAPI Recommendation includes, the Generic Security Service API with the Independent Data Unit Protection extensions, the Generic Crypto Service API, and Cryptoki. A second edition of the NSA CAPI Recommendation was release in July 96. The major changes were updates to the GSS/IDUP and GCS-API specifications and the inclusion of Microsoft's CAPI, CryptoAPI. NSA is currently validating their recommendation by developing prototype implementations.

CMAPI: Now that everyone can access cryptography, the need for accessing and utilizing certificates has become important. A subset of the NSA CAPI team authored a draft specification addressing a CMAPI. The CMAPI is composed of five sets of functions, high-level, low-level, data handling and encoding, cache management, and directory services.

KMAPI: The development of a Key Management API is just beginning. The key management services that fall within the bounds of a KMAPI include, creation, destroying, storage, data recovery, protection, and distribution of cryptographic keying material.

PANELIST: Amy B. Reiss is a computer scientist at the National Security Agency and is currently in the INFOSEC Research and Technology Office. Besides leading the in-house team on CAPIs, she is a member in the IEEE 802.10 working group on Standards for Interoperable LAN/MAN Security (SILS). She was also a member of the Data Communications Protocol Standards (DCPS) Technical Management Panel (DTMP) working group on security and the OSE Implementor's Workshop (OIW) Security Special Interest Group. She received a B.S. in Computer Information Science Engineering at the University of Florida and a M.S. in Computer Science Engineering at Loyola College in Baltimore, where she was a member of Upsilon Pi Epsilon (UPE).

FORTEZZA and CAPIs: NSA is investigating the use of commercial Cryptographic Application Program Interface (CAPI) standards for use with the FORTEZZA crypto card. Most developers of FORTEZZA-enabled applications communicate to the card through a very low-level CAPI, known as the CI Library, that was designed especially for the FORTEZZA card. The CI Library gives the developer maximum flexibility in handling the card, but requires the developer to have considerable knowledge of both cryptography and the FORTEZZA card. NSA is modifying the CI Library to a more abstract commercial CAPI, such as Cryptoki. NSA will also investigate high-level CAPIs, such as GCS-API and GSS-API.

PANELIST: John Centafont works at the National Security Agency and hold a B.S. in Electrical Engineering from Drexel University and a M.S. degree from the Johns Hopkins University. He is currently responsible for the development of the specifications and software products that allow application developers to integrate the FORTEZZA Crypto Card. This effort involves the incorporation of PC Card standards, security protocols, and commercial CAPI developments.

The Microsoft Internet/Intranet Security Framework: Microsoft will provide an overview of the Microsoft Internet Security Framework - a comprehensive set of public-key and password-based security technologies that give you the ability to securely exchange information across public and private networks, control access from public networks to private networks, and engage in electronic commerce. Topics will include cryptographic APIs, secure channel communications, certificate management, digital signatures, C2 security, network security, and electronic commerce.

PANELIST: TBA

Public Key Infrastructure:

PANELIST: Lawrence G. Dobranski is the Canadian Communications Security Establishment's Manager of ITS Industrial Programs, Standards and Initiatives. Before being appointed to his current position he was the INFOSEC Liaison officer for CSE to the National Security Agency and the National Institute of Standards and Technology in the U.S. He has a varied ITS background in both policy and technical areas.

He chaired the information technology working group during the core list exercises of the Coordinating Committee on Multilateral Export Control (CoCOM) held in Paris from the fall of 1990 through the spring of 1991.

Lawrence has held several ITS and computer/communications engineering positions at CSE. His work has involved Fiber Optic Data Distribution, INFOSEC Software Engineering Standards, and ISDN. He started his career at I.P. Sharp Associates working in real-time data acquisition and control and computer aided dispatch systems.

He has a Masters in Engineering from Queen's University and a Bachelor of Science from Dalhousie University both in Engineering-Physics.

Lawrence is an active amateur radio operator, holding call sign VA3LGD.

International Cryptography Experiment (ICE) Update:

The overall purpose of the ICE project is to develop modular, removable, replaceable cryptographic-based security components that are commercially available, satisfy a wide range of user needs, and can be easily implemented in industry testbeds. One goal of the ICE project is to establish a controlled set of experiments to test the following general hypothesis: an application that uses cryptographic based security services can be separated from specific implementations of the cryptography in such a manner that the application can use any of a set of alternative cryptographic implementations (hardware or software) without any changes within the application code. Conversely, a cryptographic implementation (generically called a token) held by one person should support all the applications performed by that person. Such an approach yields flexible combinations of security that provide a variety of protection levels. When properly integrated in the fabric of a network itself, this approach can provide sufficient robustness to a network that provides some of the features needed for survivability in the face of intentional threats of disruption.

TIS has specified an architecture for the demonstrations to be implemented as a part of the ICE project. The architecture depicts a set of logical layers between a set of information processing applications and a set of cryptographic-service providing tokens. Based on the demonstration architecture, TIS selected a set of three specific profiles for implementation and performance. Each demonstration integrates an application with a number of cryptographic modules embodying different algorithm families and optionally a key escrow technique. TIS selected the

demonstrations to achieve the overall goals of ICE, including learning how to use multiple cryptographic mechanisms in multiple applications, both easily and securely. This information includes how to specify, implement, and effectively use modular, removable, replaceable components in the CAPI
architecture. Satisfying varying protection requirements in diverse environments (e.g., military, non- military government, domestic commercial, multi-national commercial, individual) without impacting applications is included. Providing robustness or survivability through rapid deployment of alternative components without requiring reengineering or even field modifications is also included.

This panel presentation will quickly review the background and motivation for ICE, review our current plans, and provide current status and results, focusing on the first of three demonstrations.

PANELIST: David Balenson is a Principal Computer Scientist in the Advanced Research and Engineering (AR&E) Division at Trusted Information Systems where he participates in assorted projects involving the design, analysis, implementation, and/or testing of Information Security (INFOSEC) systems employing embedded cryptographic-based Communications Security (COMSEC) solutions. Mr. Balenson is currently leading several cryptographic research efforts including the International Cryptography Experiment (ICE), the Worldwide Cryptographic Products Survey, and FORTEZZA integration for the TIS Gauntlet Internet Firewall. Mr. Balenson is a member of the Internet Privacy and Security Research Group (PSRG) which developed the Privacy Enhanced Mail (PEM) specifications and is involved in the ongoing design and analysis of security for other Internet protocols.

Mr. Balenson is an associate professorial lecturer at George Washington University (GWU) where he teaches cryptographic-based network security techniques and protocols.

Mr. Balenson worked for 4 years at National Institute of Standards and Technology (formerly the National Bureau of Standards) where he participated in the development of Federal and commercial computer security standards and in the research, design and development of new and advanced methods and techniques for cryptographic-based security.

ARE CRYPTOSYSTEMS REALLY UNBREAKABLE?

Panel Chair
Dorothy E. Denning
Georgetown University, Computer Science Department
225 Reiss Science Building, Washington, DC 20057-1232

Panelists
Steven M. Bellovin
AT&T Research
600 Mountain Avenue., Murray Hill, NJ 07974

Paul Kocher
Independent Cryptography Consultant
P.O. Box 8243, Stanford, CA 94309

Arjen K. Lenstra
Citibank
4 Sylvan Way, Parsippany, NJ 07054

Eric Thompson
AccessData Corporation
560 South State Street, Suite J-1, Orem, UT 84058

Panel Summary

We often hear the claim that today's codes are unbreakable. But are they, their implementations, or the systems that use them really secure? This session will explore the strength of existing systems in terms of potential weaknesses in algorithms, protocols, implementation, and application environments. Speakers will explore mathematically secure designs vs. systems that are secure in practice and measures for quantifying security. Recent efforts in factoring, code breaking, and vulnerability analysis will be discussed, along with what developers and users can do to improve security.

THE MATHEMATICAL PRIMITIVES: ARE THEY REALLY SECURE?

Arjen K. Lenstra
Citibank
4 Sylvan Way, Parsippany, NJ 07054

Panel Statement

Corporations are beginning to see that venturing out on the Internet with a homepage on the web is to increase visibility and to draw attention. Unfortunately the audience includes not only potential customers but also virtually all hackers worldwide. At least some of them will, intentionally or not, cause trouble.

Solutions to the resulting security problems are not hard to find on the net, since many software vendors now advertise "secure" versions of their products. This makes using the net really risky, because users might mistakenly believe they are well protected. The widely publicized and rather frequent news stories about network break-ins and imperfections in security software should dispel such illusions. It seems that our competence to secure the net cannot keep up with our desire to use it.

Despite the confusing array of security solutions, there are only a few mathematical primitives on which they are based. Even in faulty security products, the soundness of the underlying mathematics is hardly ever in question; it is the way it is used that causes the vulnerabilities. In this presentation I discuss the mathematical primitives, not the many slippery ways in which they are employed. I concentrate on the primitives themselves, the assumption of their soundness and will discuss the latest theoretical and practical developments.

New Paradigms for Internetwork Security

J. T. Haigh

Secure Computing Corporation
2675 Long Lake Road
Roseville, MN 55113

Each year the New Security Paradigms Workshop provides researchers with an informal environment in which to discuss new approaches to security with their peers. As such, it provides an excellent opportunity for feedback at an early stage of the work. This year, as in previous years, the Workshop has organized a panel based on some of the more interesting concepts presented at the Workshop. One very strong theme at this year's Workshop was the need to identify new approaches for providing security in very heterogenous, highly internetworked environments.

Each of the participants on this panel writes of that need and proposes approaches for addressing it. In his paper, "The Emperor's Old Armor," Bob Blakley, from IBM in Austin, Texas, paints a grim picture of the current state of computer security and suggests a new set of foundational assumptions. In the paper, "Reactive Security and Social Control, written by Sverker Janson and his associates from the Swedish Institute of Computer Science, we find an argument for what they call soft security mechanisms, such as runtime monitoring and control of programs based on their expected behaviour. Steven Greenwald, from NRL, suggests that we use Role Based Access Control policies to put more control of resources in the hands of individual users. Finally, William Wulf and his colleagues describe the security model for Legion, a highly heterogeneous distributed system that they intend to prototype at the University of Virginia.

As this note goes to press, we are planning on the following format, which is subject to change. After each panelist has presented on overview of his position, they will discuss each other's positions. This will be followed by a broader discussion involving the audience. Following a short break, we will have the room for another hour or so to allow for unstructured discussion among the panelists and the audience.

THE EMPEROR'S OLD ARMOR

Bob Blakley
blakley@vnet.ibm.com

The traditional computer security model is built around a "reference monitor", supported by hardware protection mechanisms, which enforces administratively defined security policies. The reference monitor's software is assumed to be of high reliability and integrity. The reference monitor is supplemented by strong cryptography for those unfortunate moments when our data must venture outside the cozy confines of its safe haven.

This model's analogies are mostly military: the image is that of an *information fortress*, with walls, guards, interior compartments, and a defending army. When you approach the information fortress's outer wall ("security perimeter"), you present your "password" to the guardian of the gate. The fortress's defensive garrison ("access control" facilities) protect your "confidential data" until you want to send it out of the "security perimeter", perhaps through a "firewall", at which point you use a code (but only in your home country -- because cryptography is a "munition"!) The system's strong walls and trustworthy gate guards ("integrity features of the Trusted Computing Base") protect it against the introduction of "Trojan Horses" and "logic bombs".

The information fortress model was designed for (and in) a world in which computers were expensive, solitary, heavy, and rare. But that world is long gone. Information fortresses are not protecting today's information much more effectively than Europe's magnificent physical fortresses are protecting today's national borders.

The state of computer security is dismal. The same exposures keep recurring; we make no practically useful progress on the hard problems of integrity, assurance, policy, and interoperability; and we are less and less able to adapt the fortress model to new technologies as they arise. Computers are rapidly getting smaller, cheaper, and more richly connected. More and more data resides on machines incapable of meaningful physical security (for example, laptop computers and "personal digital assistants") and designed -- by economic necessity -- with no strong logical security. Even the relatively few remaining information fortresses have thrown open their gates to Ethernet, ISDN, and fiber connections. At the other end of those connections lies the worldwide Internet, on which, as Steve Bellovin has observed, "There Be Dragons".

Technologies more disruptive than the Internet loom on the horizon; object-orientation blurs the distinction between data and code, robbing us of one of our most powerful integrity tools (hardware-enforced memory protection). At th .
e same time object orientation encourages us to "reuse" code written by others -- in some cases without benefit of access to the source text of the code we reuse. "Intelligent Agent" architectures invite us to execute other peoples' code on our systems and to write our own code and send it out to make its way in the world without benefit of our oversight. These agents are not distinguishable from programs which we describe as "viruses" today.

The software industry is in general not keeping up with the escalating threat; most modern software is designed without any thought given to security up-front. The Internet, OMG CORBA, the Worldwide Web, and most Personal Computer operating systems are examples of major components of the worldwide software infrastructure into which security is currently being retrofitted.

The Information Fortress model is based on three principles; the security community's dirty little secret is that all three of these principles rest on infirm foundations:

1. Policy

Policy scales poorly in every dimension. As the number of subjects authorized to use the system, the number of objects managed by the system, and semantic complexity of operations provided by the system increase, the policy administrator's job quickly spirals out of intellectual control.

2. System integrity and the reference monitor

"System integrity" assures that the security policy of a system cannot be bypassed. The US National Computer Security Center defines "integrity" as follows [NC88]:

> "sound, unimpaired, or perfect condition"

This sets the bar pretty high. But perfection really is the standard, because any hole in the wall of the fortress will let the enemy in.

Implementing a high-integrity system is prohibitively costly and difficult.

3. Secrecy

The fortress model depends heavily on secrecy. The security community has long recognized the problems associated with secrecy and has shrunk the secrecy perimeter to exclude everything except cryptographic keys; this has been formalized as Kerchoff's principle: "security is in the keys", which is intended to mean that if the keys remain confidential, the system is secure. But decades of experience with the problems of passwords and crypto key management suggest that a more accurate formulation might be "insecurity is in the keys"

The simple problem with secrets is that people are not good at keeping them. But there are also complicated problems. It is not always clear, for example, what information constitutes a secret, or what information will reveal it to a particular person.

The central proposition of the paper, therefore, is:

> No viable secure system design can be based on the principles of Policy, Integrity, and Secrecy, because in the modern world Integrity and Secrecy are not achievable and Policy is not manageable.

This is why computer security is starting to fail - and why it will continue to fail until it is re-built on new foundations. The paper urges a search for these new foundations, and suggests some guiding principles:

- Assume low integrity.
- You can't keep a secret.
- Security should be inherent, not imposed.
- Policy is evidence that security is imposed.
- Identity is a side-effect of policy (don't depend on it; don't authenticate it).
- Trust is is evidence that security is imposed (trust nothing and no one).
- Ease of use should be proportional to the probability that use is harmless.
- Make the user ask forgiveness, not permission.
- Plan for emergence.
- Privacy is not secrecy.
- Protection is not control.
- Security is not: confidentiality, integrity, availability.
- Good enough is good enough. Perfect is too good.
- Evolve!

The Emperor's Old Armor

Bob Blakley

blakley@vnet.ibm.com

October 1996

We are Losing the War

Attack Trends

- Attacks increasing
- Losses increasing
- Success rate very high
- Attack tools easily available

System Trends

- Complexity increasing
- Size increasing
- Connectivity increasing
- Price decreasing

Protection Trends

- Typical system's integrity is low and getting lower
- Assurance still expensive, difficult, slow
- Crypto still rare, heavily restricted
- Authentication, authorization, audit technology primitive, complex, and non-standard

We have the Wrong Model

The "Information Fortress"

Policy

- Scales poorly
- Administration is complicated and sensitive

Integrity

- Ultimately requires perfection; practically:
- Expensive
- Difficult
- Slows down product development

Secrecy

- People can't keep secrets
- Management and use of secrets is technically difficult
- Definition of "keeping a secret" has subtle problems

Inherent vs. Imposed Properties

Inherent properties

- Size
- Weight
- Radioactivity
- Difficulty
- Obscurity

Imposed properties

- Authorization
- Authentication
- Encryption

Security arising from inherent properties requires no management and does not require high-integrity implementations.

Security arising from imposed properties must be managed and requires high-integrity implementations.

Security should be inherent, not imposed.

Conclusions

Sun Tzu said

What is of supreme importance is to attack the enemy's strategy.

Next best is to disrupt his alliances.

Next best is to attack his army.

The worst policy is to attack walled cities.

Attack cities only when there is no alternative.

Our mistakes:

- To assume we can build walled cities around everything we care about in the information world
- To assume that the enemy will always adopt "the worst policy"

A Manifesto

No viable secure system design can be based on the "Information Fortress principles" -- Policy, Integrity, and Secrecy -- because in the modern world Integrity and Secrecy are not achievable and policy is not manageable.

THE EMPEROR'S OLD ARMOR

Inspirations

Economics

- Game theory
- Supply/demand
- Rational economic agents; self-interest; utility
- Incentives and disincentives
- insurance and actuarial science

Biology

- Immune systems
- Epidemiology
- Evolution
- Symbiosis

THE EMPEROR'S OLD ARMOR

Principles

1. Assume and accommodate low integrity
2. You can't keep a secret
3. Security should be inherent, not imposed
4. Policy is evidence that security is imposed
5. Identity is a side-effect of policy (don't use it; don't authenticate it)
6. Trust is evidence that security is imposed (don't use it)
7. Ease of use should be proportional to the probability that use is harmless
8. Make the user ask forgiveness, not permission
9. Plan for emergence
10. Privacy is not secrecy
11. Protection is not control
12. Security is not: confidentiality, integrity, availability
13. Good enough is good enough. Perfect is too good
14. Evolve!

THE EMPEROR'S OLD ARMOR

Position Statement for New Paradigms for Internetwork Security Panel

Steven J. Greenwald
Email: greenwald@itd.nrl.navy.mil
WWW: http://www.itd.nrl.navy.mil/ITD/5540

Center for High Assurance Computer Systems
Naval Research Laboratory
Washington, DC 20375
United States of America

Introduction

The security policy currently used on most distributed systems is an old one, dating back to simpler times when most computer systems were centralized. This security policy is based on the idea that there is a central managing authority, called the *system administration*, that is ultimately responsible for the management of computer security within an administrative domain. In this security policy system administration includes the management of system resources, user accounts, and user privileges. This security policy is typified by an operating system such as UNIX. I refer to this older security policy as the *Jurassic Age Security Policy* (JASP) since it apparently dates back to the time when huge dinosaur computers were kept in air-conditioned pens, lazily grazing on their data, before faster, leaner machines wiped them out.[1]

JASP introduces difficulties when working in a distributed computing environment, and most of the computer systems in use on the Internet are based on JASP. I am specifically concerned with the management of system resources and access control in a distributed computing environment. We need a new paradigm for security that is congruent with the highly distributed nature of the Internet.

Paradigm Problems

JASP presents the following problems when working in a distributed environment.

1. User-names are often duplicated across name-space domains in a distributed system. For example, two different users may have the same user-name on two different hosts within a distributed system.

[1] I am obviously open to suggestions for more appropriate terminology, and I'm also interested in exactly when JASP first came into existence.

2. Location transparency may not be possible. For mobile users who often change hosts, the combination of user-name and host-identifier fails to uniquely identify the user. One user may have two (or more) different user-names at different locations. Two users in different administrative domains may have the same user-name.

3. There exists a "weak link in the chain" effect. The security of the entire distributed system depends upon the security of the individual hosts that are being used within a group of administrative domains. One lax system administration can compromise an entire distributed system.

4. Users often need to assume different roles, and JASP does not accommodate this. I define a role as a labeled set of capabilities that a user can activate. Roles, as opposed to protection groups, are generally considered to be a form of mandatory access control. For example, a user may wish to simultaneously assume the roles of "panelist" and "chair" for a particular session.

5. It is often very difficult to share resources with other users on other computer systems without getting permission from the system administrations involved. Especially for real-time applications.

6. Foreign user accounts are often necessary to correct the previous problem. This places a management burden on the system administration and there is the very serious difficulty of the system administration initially verifying the identity of these foreign users. In addition, foreign user accounts present the potential problem of giving the foreign user too many permissions.

7. Military chain of command systems and corporate hierarchical systems may be difficult to model and implement because their structure clashes with the "flat" structure of the omnipotent-system-administrator approach of JASP.

Solution Requirements?

There are many ways to solve the above stated problem. I believe the best solution will contain elements of a libertarian (classical liberal) philosophy that maximizes the freedom of users while limiting system administration intervention to only vitally necessary functions. Philosophically, this should have the benefits of allowing users as much flexibility in managing their affairs as possible, while eliminating much of the drudgery commonly associated with system administration. I believe a this approach is a good compromise between authoritarian control and anarchy. I believe this because of the common observation that the Internet is the closest thing to a workable, successful anarchy that the modern world has ever developed. Yet it is this very anarchy that is now causing our present security concerns.

In a libertarian approach, users would be give more power than they currently have. This does not mean that system administrators need give up any of their power or control. In fact, it will probably mean that system administrators will be giving up the things they commonly associate with drudgery.

Since user processes and resources are for all intents now decentralized in many distributed systems, it makes sense to decentralize the method of access control, and the method of resource management.

First, we can do away with the requirement that applications identify users by operating system dependent user names and paths. I believe that role based access control (RBAC) is the preferred way to solve this problem. At a minimum, a role would need to be composed of a label (name), a set of capabilities, and a list of users that are members of that role. In addition, roles can be designed to be related to users in a many-to-many way, so that users can effectively share the same role (many users to one role) and individual users can be members of more than one role (many roles for one user). If required, auditing of users can still take place, even at the operating system level.

With RBAC, we gain several advantages. The name of the role can be more descriptive than often cryptic user names, anonymity is possible, the many-to-many relationship allows users to assume different roles, and more than one user to use the same role. The management of roles becomes part of the particular distributed application, instead of an operating system dependent issue. With RBAC, system administrators would not be pestered with user requests for foreign accounts, requests to add users to protection groups, and so forth, since these functions can be handled by users activating other roles.

Resource management is the other area where our solution lies. Currently, all resources are, in some sense, "owned" by the administrative domain they belong to. This is the wrong paradigm to use in today's decentralized world. Looking at this from a libertarian point of view, it would be better if users could logically "own" the resources they have been allocated, and deal with them as they see fit (in a secure way, of course), allowing for things such as n-person rules, different decision support mechanisms, and so forth.

For example, if a user has a certain amount of storage space allocated, why can't that user let other users access that storage space, without having to pester a system administrator? This is a common problem in systems such as UNIX, where only someone with the highest permissions can add someone to a protection group. It makes more sense to allow individual users to perform these functions, since they have already made the decision.

In addition, there should be no reason to logically view these resources as belonging to particular centralized machines. Users should be allowed to logically share their resources across administrative domain boundaries, and use them as they see fit (*e.g.,* in collaborative ways such as multiple authors writing a paper in real-time).

Utilities and security policies can and should be designed to accommodate these necessary elements. Some of the issues to be solved in these policies are things such as the exact mechanism of RBAC, how to manage resources efficiently across administrative domains, how to handle the name-space that will occur with such systems, and how to organize the combination of RBAC and distributed resource management in a coherent manner that users can understand and use.

But the most important goal of all is that we must free users from a large amount of dependence on various administrative domains, while simultaneously freeing the various system administrations from many tedious tasks. I believe that this point will become increasingly important as distributed systems continue to multiply.

Reactive Security and Social Control

Lars Rasmusson, Andreas Rasmusson, Sverker Janson *
Swedish Institute of Computer Science

Untrusted Code

A major security problem for a network oriented environment is executing untrusted code. Private information risks being disclosed or tampered with if unverified remote code manages to gain access to local resources. Since a program only shows to the user what it wants the user to see, it can hide some of its actual actions. This is the essence of a *Trojan horse*.

In the days before global networking the acts of malicious programs mainly perpetrated random acts of vandalism, like erasing files. Now, as computers get increasingly connected, programs can communicate back to their creators. This enables a new range of crimes.

As we begin to use open computer networks to transfer information of more direct economic value, we'll find that programs can to do more malicious things than erasing files. Viruses can be used to snoop passwords to valuable information services or getting hold of e-cash stored on our hard disks. As the trend leads towards where we are down-loading and executing many new programs every day, these problems are only likely to increase.

Certifying every program on the Internet would hinder the introduction of new programs, services and even of bug-fixes. The essence of the open net is that new information is put there almost instantaneously. Hence we can't do away with the concept of an untrusted program.

Cryptographic methods like digital signatures can be used to authenticate the sender and/or guarantee that the program hasn't been tampered with. However, the program's hostility cannot be decided by any level of cryptography.

For the untrusted software to be useful it may have to be granted access to information that it potentially can misuse. There is a notion of *risk* involved in dealing with untrusted code, and this is not well supported in conventional computer security. Two ways to deal with the risks are

- to use a system where trust/distrust is an integral part of the system
- runtime monitoring of the untrusted code and decision support for the user

These methods belong to a class of security mechanisms we call *soft* security. Soft security, as opposed to *hard*, means that privileges are granted as they are needed, with the current risks taken into consideration. Hard security denotes methods that don't reevaluate granted privileges.

Soft security is related to reactive, "after-the-fact", security and intrusion detection. The term reactive emphasizes the *when* the analysis is done whereas *soft* is an indication of on what grounds resource access is granted, hence the new term.

Why are there malicious programs?

Some crimes (like occasional speeding, or some white collar crimes) can be said to be *rational* in a game theoretic sense. This means that the expected net payoff is greater than zero, after considering risk of being caught, expected punishment and expected gain [1]. If we radically change the properties of an economic system (as ours will be changed by Internet) we might find that a number of new crimes will be economically "sound."

But apart from programs written in evil spite, like Trojan horses or viruses, a program can also start to misbehave because of bugs or because it is being used in a context for which it was not designed. An interconnected ever-changing program environment makes it virtually impossible

*Email: {lra, ara, sverker}@sics.se

for the programmer to understand all the effects of his/her program. Therefore it seems wise to treat all programs with a little caution, regardless of the author's intentions.

Reputation and anonymity instead of blind trust

One way to help a user to minimize the risk of using untrusted software is to use reputation mechanisms [2]. Reputation enables us to dare to take larger risks with the programs we believe are benevolent. Microsoft's reputation makes us dare to install the annual version of MS Word without first verifying its source code. But for Internet systems dealing with lots of untrusted code reputation mechanisms need to be made explicit.

The importance of being able to trust one's business partner is beginning to be acknowledged on the Internet. Certification companies and authorities that act as *Trusted Third Parts* are proliferating on the net. A trusted third part acts as a guarantor for the seriousness of the other part. However, authoritative trust has some drawbacks. It is centralized and hierarchical and it puts both parties in the hands of the trusted third. It ends up in a circular reasoning; "How do I trust the trusted part?"

What we are looking for is a system where all parties actively cooperate to build up reputation, and where reputation is built on rational grounds. Further, it should not be necessary to keep global registers over every person on the Internet, since it is both impossible and violates personal integrity.

Anonymity

Part of tomorrow's business will be conducted by programs. Unlike ordinary companies or persons, a program does not have any physical manifestation that guarantees that it will be around for a longer time. It can be copied infinitely, or changed into unrecognizability. If a program put on the Internet cannot be traced back to an originator, it is effectively an anonymous program. No-one can be held responsible for its actions. With complete anonymity, selling stolen information or goods, computer break-ins without risk of being caught, or plain vandalism (making other peoples computers crash, etc.) can be safely committed.

The converse, complete identification in all steps, is also susceptible to new crimes. Complete logging of someone can generate a computer shadow that could be used for annoying advertising or blackmailing, for break-ins ("locate persons who have bought a new VCR and who are at work") etc.

To anonymity, two approaches are possible. Either say "let's just forbid anonymous programs - everything must be traceable back to the originator," a common view. Or say "we can't prevent anonymous programs - we must therefore design our system so that it doesn't collapse if anonymous programs slip in."

The former view is unacceptable since there is no way to "forbid" some programs in an open network such as Internet to which anyone can connect their computer. If we insisted and designed such a system, it would be very vulnerable if someone released such a program in the system. It's not a good idea to construct open systems so they only work if all the other components work as intended.

Reputation demands identity

In a reputation based system identity is something valuable. Reputation coupled to an identity enables two parties to make business together, something from which they both benefit. Loosing one's reputation equals a loss of income from other business. It becomes irrational and costly to waste one's reputation by malicious behaviour.

Unforgeable identities don't have to be created by an identity issuing authority. They can be created by anyone using digital signatures, or zero-knowledge (interactive) protocols. The id works as an unforgeable trade mark, and reputation is established when the same id is used more than once. Anonymity is achieved by creating a new id for every transaction.

If we manage to construct a system where the interacting programs are "suspicious" to one another and don't expect other programs to behave nicer than their reputation guarantees, we will in fact have a system that can support anonymous programs. It will not have drawbacks such as complete logging of every person, or centralized trust servers. It acts cautiously if new (possibly buggy or malicious) components are added, but once they have merited a certain amount of trust, they are integrated into the system.

Behavior-based resource granting

Traditionally operating systems limit user access to system resources by enforcing access rules in system calls. Ordinary read/write access control and *capability systems* are examples of this. Once granted permission, the program has free access to the resource.

These security barriers are necessary but not sufficient in a networked environment where programs are exchanged promiscuously between computers. Capability systems do not solve the problems of denial-of-service attacks or of leaking information, since they are just a means to decrease granularity for the privilege assignment.

Many useful restrictions are not possible to enforce before runtime. For instance, forbidding simultaneous access to a shared resource by two or more programs inhibits potential covert channels between the programs. Whether covert communication will take place or what information might be leaked depends on the actual situation. It probably doesn't matter as much if your word processor leaks the contents of your shopping list as if it leaks your business mail.

Assigning correct privileges to a program requires the user's afterthought and skill and will be very cumbersome as the number of programs we interact with each day increases. Instead of just classifying the resources, untrusted programs could be classified by their expected behavior. This would constrain the range of expected "normal" actions and making it harder for a program to undetected do something unexpected/malicious. We are studying how to give automated support for deciding if and how a program deviates from its expected behaviour [3].

Different implementations of a service could, if similar enough, be classified as belonging to the same class. Behaviour classes reduce the number of choices the user has to make since the user needn't be aware of all rules a particular behavior implies. Since a violated rule can be explained in behavioral terms it is easy to understand and giving the user decision-support for how to handle the situation.

Automatically communicating and updating behaviour descriptions, will make the society of Internet hosts more resilient to malicious programs.

Conclusions

We need to remove the obstacles for an open Internet with commercial interests without using centralized solutions. Trust relationships could be used to assess the economical risks of engaging in activities with unknown parties. To inhibit malicious programs from covert activities we suggest runtime monitoring for constraining the allowed behavior of programs.

References

[1] Economic Times *The Economics of Crime*, journal, vol 4, no.1, 1995. Addison-Wesley.

[2] Rasmusson, Lars & Janson, Sverker *Simulated Social Control for Secure Internet Commerce* New Security Paradigms Workshop '96, 1996.

[3] Rasmusson, Andreas & Janson, Sverker *Personal Security Assistant for Secure Internet Commerce* New Security Paradigms Workshop '96, 1996.

NISS Whitepaper: A New Model of Security for Distributed Systems

Wm A. Wulf, Chenxi Wang, Darrell Kienzle
University of Virginia

Given the brevity of this paper, we will primarily present a problem and only hint at our approach to solving it.

The conventional security approach has been for "the system" to mediate all interactions between users and resources, and to enforce a single system-wide policy. This approach has served us well in the environment of a centralized system because the operating system implements all the key components and knows who is responsible for each process. Alas, it simply will not work in large, open, distributed systems. Thus, the authors are investigating a new model of computer security in the context of a new distributed system, Legion, being built at the University of Virginia.

Users of Legion-like systems must feel confident that the privacy and integrity of their data will not be compromised — either by granting others access to their system, or by running their own programs on an unknown remote computer. Creating that confidence is an especially challenging problem for a number of reasons; for example:

- We envision Legion as a *very* large distributed system; at least for purposes of design, it is useful to think of it as running on millions of processors distributed throughout the galaxy.
- Legion will run *on top of* a variety of host operating systems; it will not have control of the hardware or operating system on which it runs.
- There won't be a single organization or person that "owns" all of the systems involved. Thus no one can be trusted to enforce security standards on them; indeed, some individual owners might be malicious.

No single security policy will satisfy all users of a huge system — different individuals and organizations will have different views of what is necessary and appropriate. We cannot even presume a single "login" mechanism — some situations will demand a far more rigorous one than others. Moreover we cannot anticipate all the policies or login mechanisms that will emerge; both will be added dynamically. And, for both logical and performance reasons, the potential size and scope of Legion suggests that we should not have distinguished "trusted" components that could become points of failure/penetration or bottlenecks.

Running "on top of" host operating systems has many implications, but in particular it means that in addition to the usual assumption of insecure communication, we must assume that copies of the Legion system itself will be corrupted (rogue Legionnaires), that some other agent may try to impersonate Legion, and that a person with "root" privileges to a component system can modify the bits arbitrarily.

The assumption of "no owner" and wide distribution exacerbates these issues, of course. Since Legion cannot replace existing host operating systems, the idea of securing them all is not a feasible option. We have to presume that at least some of the hosts in the system will be compromised, and may even be malicious.

These problems are sufficiently different from those faced by single-host systems that some of the assumptions that have pervaded work on computer security must be re-examined. Consider just two such assumptions. The first is that security is absolute; a system is either secure or it is not. A second is that "the system" is the enforcer of security.

In the physical world, security is never absolute. Some safes are better than others, but none is expected to withstand an arbitrary attack. In fact, safes are rated by the time they resist particular attacks. If a particular safe isn't good enough, its owner has the responsibility to get a better one, hire a guard, string an electric fence, or whatever. It isn't "the system that provides added security.

We said that users must feel "confident"; we did not say that they had to be "guaranteed" of anything. Security needs to be "good enough" for a particular circumstance. Of course, what's good enough in one case may not be in another — so we need a mechanism that first lets the user know how much confidence they are justified in having, and second provides an avenue for gaining more when required.

The phrase "the trusted computing base" (TCB) is used to refer to systems that enforce a security policy. The mental image is that the TCB mediates all interactions between users and resources, and for each interaction decides to permit or prohibit it. Even communications, which is inherently insecure, is usually presumed to be inside the TCB perimeter and the system is considered to be responsible for implementing secure communication on top of the insecure base.

As with the previous assumption, this one just doesn't work in a Legion-like context. In the first place there isn't a single policy, new ones may emerge all the time, and the complexities of overlapping/intersecting security domains blur the very notion of a perimeter to be protected. In the second place, since we have to presume that the code might be reverse-engineered and modified, we cannot rely on the system enforcing security!

Moreover, security has a cost in time, convenience, or both. The intuitive determination of how much confidence is "good enough" is moderated by cost considerations. As has been observed many times, one reason that extant computer systems have not paid more attention to security is that the cost, especially in convenience, is too high. These prior systems took the "security is absolute" approach, and everyone paid the cost regardless of their individual needs. To succeed, our model must scale — it must have essentially zero cost if no security is needed, and the cost must increase in proportion to the extra confidence one gains.

The above observations call for rethinking some very basic, often unstated assumptions. In the rest of the paper, we suggest a new security model for Legion. The model, responds to the issues raised above; its premise is that we cannot, and indeed should not, provide a *guarantee* of security. What we can and should do is (1) be as precise as possible about the degree of confidence a user can have, (2) make that confidence "good enough" and "cheap enough" for an interestingly large selection of users, and (3) provide a context that allows the user to gain the additional confidence they require with a cost that is intuitively proportional to the added confidence they get. The model is derived from three principles:

- First, as in the Hippocratic Oath, *do no harm*! Legion's first responsibility is to minimize the possibility that it will provide an avenue via which an intruder can do mischief to a remote system.
- Second, *caveat emptor*, let the buyer beware. In the final analysis users are responsible for their own security. Legion provides a model and mechanism that make it feasible, concep-

tually simple, and inexpensive in the default case, but in the end the user has the ultimate responsibility to determine what policy is to be enforced and how vigorous that enforcement will be.

- Third, *small is beautiful.* That is, given that one cannot absolutely, unconditionally depend on Legion to enforce security, there is no reason to invest it with elaborate mechanisms. On the contrary, at least intuitively, the simpler the model and the less it does, the lower the probability that a corrupted version can do harm.

Legion is an object-oriented system. Thus, to implement these principles

- the unit of protection is the object, and
- the "rights" to the object are to invoke its member functions (each member function is associated with a distinct right).

This is not a new idea; it dates to at least the Hydra system in the mid 1970's; what is somewhat more novel is the way rights are enforced. In line with the "small is beautiful" principle, there are just four basic concepts to the enforcement mechanism:

- every object must provide certain member functions (that may be defaulted to NIL);
- there is a "responsible agent" (RA) associated with each operation. User-defined objects play the role of RA by supplying an appropriate set of member functions.
- every invocation of a member function is performed in an environment consisting of a pair of (unique) object names — those of the operative responsible agent, and "calling agent".
- there are a small set of rules for actions that Legion will take, primarily at member function invocation. The general approach is that Legion will invoke the known member functions, thus giving objects the responsibility of defining and ensuring the policy.

It's not that easy, of course. In a large distributed system it is impossible to prevent corruption of some computers. We must presume that someone will try to pose as a valid Legion system or object in order to gain access to, or tamper with other objects in an unauthorized way. On the other hand, perhaps we can make the probability of such mischief sufficiently low and its cost sufficiently high to be acceptable for all but the most sensitive applications. We are exploring a number of approaches to this, including: *Defense in depth*:, *Least Privilege*:, *No hierarchy (compartmentalize)*:, *Minimize functionality to minimize threats*:, *If it quacks like a Legion.* (that is, Legion is defined by its behavior, not its code), *Firewalls*:, and *Punishment vs. Prevention*:

The model we have posited, we believe, is both a conceptually elegant and a robust solution to the problems posed earlier. We believe it is fully distributed; it is extensible to new, initially unanticipated types of objects; it supports an indefinite number and range of policies and "login" mechanisms; it permits rational, user-defined trade-offs between security and performance. At the same time, we believe that it has an efficient implementation.

What we need to do now is to test the "we believe" part of the last paragraph.

Public Key Infrastructure: From Theory to Implementation

Panel Chairs: W. Timothy Polk, NIST and Donna F. Dodson, NIST

A certificate-based public key infrastructure (PKI) can provide a mechanism to establish trust relationships and obtain security services. The trust relationships may transcend organizational and even international boundaries, even if the parties were previously unknown to each other. The security services supported can include integrity, confidentiality, and non-repudiation. While the technical promise of a PKI is clear, the corresponding operational issues are not as well understood. The purpose of this session is to provide an in-depth view of the issues involved in implementing and maintaining a public key infrastructure.

To support security services on a broad scale for government or industry, a PKI is an appropriate vehicle. However, implementing and maintaining a PKI is unfamiliar territory. How does an agency or company develop a PKI that will support its internal security requirements today and be positioned to integrate with external PKIs as they emerge?

Recent developments provide valuable insight into these questions. Maturing technical specifications should provide future interoperability. Pilot projects have been performed and initial implementations of PKIs are being developed for various branches of the federal government. The Canadian government is currently implementing their own PKI. The lessons learned in these projects can guide others in the implementation of their own PKIs.

The purpose of this panel is to familiarize the audience with standards, interoperability, and implementation issues. Panel members will discuss relevant technical specifications, security policies for PKI supported applications and PKI components, and lessons learned from pilots and current implementations.

This panel may be of interest to parties in both the private and public sectors. This includes project managers, application developers, and security officers in federal agencies and industry who are considering public key infrastructure to support their applications. This panel will be presented in two sessions: Public Key Infrastructure Technology, and Public Key Infrastructure Implementations.

Public Key Infrastructure Technology

Donna Dodson (NIST), Session Chair

- *An Introduction to Public Key Infrastructure Technology*: Russ Housley, Spyrus
- *Requirements for Digital Signatures and Supporting Services for Financial Applications*: Chris Martin, General Accounting Office
- *An Overview of Public Key Infrastructure Standards*: Warwick Ford, Independent Consultant
- *Minimum Interoperability Specifications for PKI Components*: W. Timothy Polk, National Institute of Standards and Technology

- *Security Considerations When Using X.509 Certificates*: Santosh Chokhani, Cygnacom Solutions, Inc.
- *Linking Digital Signatures with Manual Signatures*: Victor Hampel, Hampel Consulting

Public Key Infrastructure Implementations

W. Timothy Polk (NIST), Session Chair

- *Federal Public Key Infrastructure Activities*: Patricia N. Edfors, Government Information Technology Services (GITS) Working Group
- *The MISSI Rollout: Lessons Learned.* Donald R. Heckman, National Security Agency
- *NIST Implementation Projects*: Donna Dodson, National Institute of Standards and Technology
- *Security Infrastructure Program Management Office*: Richard Kemp, General Services Administration SI-PMO
- *CommerceNet Security Showcase*: James Galvin, CommerceNet
- *The Canadian Government PKI*: Wynn Redden, Communications Security Establishment

Establishing an Enterprise Virus Response Program

Tutorial to address the practical aspects of establishing a proactive response to computer virus incidents.

Provided by

Mitretek Systems
7525 Colshire Drive
McLean, VA 22102

ABSTRACT

Enterprise Virus Response concentrates on the practical issues that need to be addressed to effectively and efficiently prevent and manage computer virus incidents. Virus prevention, detection, response and tracking are important components of an enterprise response. The goal of this tutorial is to provide practical information that can be used to understand the virus threat; to institute low cost preventive mechanisms; to develop and implement enterprise response mechanisms, including when to contact the experts; and to monitor the effectiveness of the tools and program within the enterprise.

Keywords: Enterprise; Virus Response; Virus

Introduction

Recent statistics have shown that computer virus incidents continue to be a fact of corporate life. Computer viruses and other malicious code pose threats to integrity and availability, such as denial of service. The source of these threats has expanded from infected diskettes to electronic message attachments and files downloaded from the Internet. The sources and the threats are concerns for any computer user. The effort necessary to control these threats can inundate the individual but, if not done, the ramifications of virus recovery can be devastating to the enterprise. For this reason, this tutorial aims to provide practical information that can be used to understand the virus threat; to institute low cost preventive mechanisms; to develop and implement enterprise response mechanisms, including when to contact the experts; and to monitor the effectiveness of the tools and program within the enterprise.

Understanding the Virus Threat

An organization initiating an Enterprise Virus Response program should be aware of the risks, exposures, and methods of virus infection.. Awareness of virus incident characteristics, such as the types of viruses and distinctions between the common infection mechanisms, in addition to current virus trends, such as the sources of virus infections, provide insight for determining the best approach for virus prevention, detection, management, and response in a given environment.

Instituting Preventive Mechanisms

Viruses can damage not only data, but also productivity and client credibility. Such damage can be difficult to counteract. The adage that "an ounce of prevention is worth a pound of cure" fits the computer virus arena as well as the health arena. Much like a person, exposing an enterprise to a virus does not necessarily result in an infection. However, the exposure does provide an opportunity for the infection to spread. It is important, therefore, to identify the sources of exposures and the means to prevent and react to them.

To identify the sources of exposure to computer viruses, the enterprise must rely on its understanding of its own operational environment as well as the virus sources and infection mechanisms. Factors such as the operating system and networking options, business applications, operational policies, security practices, and impact on user productivity determine the appropriate prevention mechanisms. Instituting effective preventive mechanisms can, in fact, eliminate a large percentage of the threat.

Developing Enterprise Virus Response Mechanisms

Enterprise Virus Response is designed to help the organization develop a proactive program for the prevention, detection, containment, management, and recovery of computer virus incidents. This tutorial will cover the processes needed to prepare for an infection or incident, to detect and contain a virus exposure or infection, to recover from an infection, and to manage the response program.

PREPARATION

Preparation for virus incident management includes the development and enforcement of anti-virus policy, the deployment and installation of anti-virus software, and the implementation of the preventive mechanisms.

DETECTION

Despite good prevention techniques, computer virus infections and incidents still occur. The detection process includes:

- Using the anti-virus product
- Taking a sample of the virus
- Identifying the virus
- Investigating the incident

RESPONSE

A crucial portion of the Enterprise Virus Response program includes the removal of the virus and recovery of the computing environment. The computing environment includes information, storage media, network connectivity, and PC user productivity. Anti-virus products and tools can be used to remove computer viruses. However, it is not always true that the product can remove all viruses. Responding to a virus incident includes:

- Removing the virus
- Contacting the experts (if necessary)
- Recovering the data, software, and operating environment

MANAGEMENT

Information, such as the prevalence and extent of computer virus infections, gathered during computer virus incidents can be used to effectively determine the best approach for virus prevention, detection, management, and response. The management of computer virus incidents includes:

- Notifying the user or customer of an infected diskette or computer
- Reporting and recording the incident and its related activities and results
- Analyzing the incident

Monitoring the Effectiveness

Monitoring the effectiveness of the program includes two activities - analyzing the trends and identifying improvement opportunities. Analyzing trends identifies high incident areas; common computer viruses within the enterprise; anti-virus tool usage; and policy compliance. Knowledge of the trends within the enterprise facilitates informed business decisions regarding future anti-virus initiatives and improvement opportunities.

Summary

To avoid the disruption and damage caused by computer virus incidents, an Enterprise Virus Response program should emphasize the importance of prevention as well as response. Prevention is the responsibility of all PC users. The primary way to prevent computer virus incidents is for the enterprise to institute, and all PC users to adhere to, safe and sensible computing practices.

Data Warehousing, Data Mining, and Security: Developments and Challenges

Dr. Bhavani Thuraisingham

The MITRE Corporation
Burlington Road
Bedford, MA 01730

ABSTRACT OF PANEL PRESENTATION

This paper is the extended abstract of the panel presentation on Data Warehousing, Data Mining, and Security to be given at the National Information Systems Security Conference in October 1996. It is a version of the invited talk presented at the Tenth IFIP Working Conference in Database Security in July 1996. It describes security considerations for two emerging technologies: data warehousing and data mining.

1. INTRODUCTION

Data warehousing and data mining are two terms that have become an essential part of data management technology. Having a data warehouse for managing the data is becoming a necessity with many enterprises. Several organizations are building their own data warehouses. Commercial database system vendors are marketing data warehousing products. In addition, some companies are specializing in developing data warehouses. The idea behind a data warehouse is that it is often cumbersome to access data from multiple and possibly heterogeneous databases. Several processing modules need to cooperate with each other for processing a query in a heterogeneous environment. Therefore, a data warehouse will bring together the essential data from these diverse data sources. This way the users need to query only the warehouse. In addition, a data warehouse also often contains information such as summary reports and aggregates that are determined by the applications using the warehouse and the types of queries posed.

A related technology, which is used to convert the data in the warehouse as well as in other databases into some useful information is data mining. That is, data mining is the process of posing a series of appropriate queries to extract information, often previously unknown, from large quantities of data in the database. Data mining technology is a combination of various other technologies including statistics, machine learning, database management, and parallel processing. Typical data mining techniques include classification, association, and sequencing. For example, data mining by association implies detecting the following pattern: whenever John travels to London, Peter also travels with him.

The developments in data warehousing and data mining technologies have resulted in additional security concerns. For example, can information be deduced from the use of various data mining tools? What are the appropriate auditing procedures for data warehouses? This presentation will discuss security issues for data warehousing and data mining. First it will describe security issues for data warehouses. In particular, security for building the warehouse, as well as querying the warehouse will be addressed. The second half of the presentation will address data mining. In particular, the security threats due to data mining, some techniques for handling these threats, as well as the use of data mining as a tool to handle security problems will be presented. In section 2 of this abstract we will give an overview of security issues for data warehousing. In section 3 we present the relationship between security and data mining. For some background information on data warehousing we refer the reader to [INMO93]. An overview of data mining is given in [IEEE93].

2. DATA WAREHOUSING AND SECURITY

As stated in the introduction, there are two aspects to data warehousing. One is building the warehouse and the other is querying the warehouse. Many of the commercial tools focus on structuring the warehouse in such a way so that query processing can be facilitated. Building the warehouse from heterogeneous data sources is in the research stage.

Research on security for integrating heterogeneous databases will contribute significantly toward exploring security for building a warehouse. For example, when integrating multiple heterogeneous databases to build a warehouse, one may have to deal with multiple security policies. A major issue here is in dealing with inconsistent policies. One needs to resolve various conflicts and generate an appropriate security policy for the warehouse. Work has been reported in [BLAU95] on security for federated database management. One needs to examine such work in developing a security policy for the warehouse. Other issues include the security impact on (1) the data model for the warehouse, (2) generating appropriate update requests on the warehouse from the updates made to the individual databases, and (3) developing the metadata for the warehouse .

There are also some important additional security considerations in building a warehouse. This is due to the fact that when integrating heterogeneous databases, one does not assume the development of a data repository whereas in the case of a warehouse, there is usually a physical data repository. An example security concern is the following. A warehouse database may give summary information. This summary information is often derived from the data in the heterogeneous databases. It is important that one does not deduce sensitive information in the heterogeneous databases from the summary information in the warehouse. Therefore, statistical database security as well as the developments on the inference and aggregation problems will also play an important role in securing the warehouse.

The previous discussion focussed on the security issues for building a warehouse from heterogeneous data sources. Security should also be maintained while the warehouse is in operation. For example, actions on the warehouse need to be audited. The question is can the traditional database auditing techniques be used for the warehouse? Other issues include the following. Should there be a special warehouse administrator and warehouse security officer? What is the relationship between the warehouse administrator / security officer and the administrators / security officers of the heterogeneous databases used to develop the warehouse? Can appropriate query modification techniques be developed for the warehouse? Should the access control rules enforced on the warehouse be taken into consideration when structuring the warehouse depending on the queries? What is the security impact on the access methods and index strategies? How can views be used as a protection mechanism for the warehouse? Research is being conducted on addressing some of these issues. For example, Stanford University's Data Warehousing project [ZHUG95] is investigating techniques for materialized views for the warehouse as well as maintaining the views as the data sources get updated. Enforcing security through views in a warehousing environment needs more work.

Many of the issues discussed here show that security for data warehousing is a combination of security for database management systems, statistical databases and integrating heterogeneous databases. More research is needed to determine the security issues specific to the warehouse before solutions for securing a warehouse can be developed.

3. DATA MINING AND SECURITY

Recently there has been much interest on exploring the relationship between data mining and security. Some preliminary ideas were discussed at the data mining special session that took place at the Ninth IFIP 11.3 Working Conference on Database Security in 1995 [LIN95]. More details on this topic have been given in [MARK96]. There are two aspects to data mining and security. One is that data mining techniques can be be applied to handle problems in intrusion detection and database auditing. In the case of auditing, the data to be mined is the large quantity of audit data. One may apply data mining tools to detect abnormal patterns. For example, suppose an employee makes an excessive number of trips to a particular country and this fact is known by posing some queries. The next query to pose is whether the employee has associations with certain people from that country. If the answer is positive, then the employee's behavior is flagged.

Current data mining tools are sufficiently advanced so that one could start applying them to detect intrusions and abnormal behavior. However, many of these tools work on structured databases such as relational databases. Therefore, the data to be examined has to be first converted to structured format so that these tools can be applied. Recently, an investigation was reported in [GRIN96] where the idea is to place network intrusion data to be mined in various repositories. This will enable researchers as well as developers to test the algorithms and tools on these common repositories to see if suspicious behavior could be determined. In other words, the network intrusion data sets to be explored (e.g. mined, visualized, etc.) will enable researchers to compare various approaches to data exploration. Data mining, visualization, and any other collection of tools as well as the human expert may be used in this process. The goal is to determine what tools can help in discovering real time suspicious behavior. Research is also beginning on data mining for unstructured data such as text and images. As developments are made, one could expect to have tools to apply on unstructured audit data.

The second aspect to data mining and security is the inference problem. That is, while the previous example shows how data mining tools can the used to detect intrusions and abnormal behavior, the next example shows how data mining tools can be applied to cause security problems. Consider a user who has the ability to apply data mining tools. This user can pose various queries and infer sensitive hypothesis. That is, the inference problem occurs via data mining. There are various ways to handle this problem. One approach is as follows. Given a database and a collection of data mining tools, apply the tools to see if sensitive information can be deduced from the unclassified information legitimately obtained. If so, then there is an inference problem. Such an approach may be carried out periodically as the database gets updated. There are some issues with this approach. One is that we are applying only a limited set of tools. In reality, the user may have several other data mining tools available to him. Furthermore, it is impossible to cover all ways that the inference problem could occur.

Another approach which is much harder to accomplish is to apply a data mining-based inference controller during run time. This means when a user poses a query, determine whether by releasing the results an inference problem could occur. The inference controller in this approach will be based on a collection of data mining techniques such as classification, association, and sequencing. For example, suppose we want to protect the fact that whenever Peter travels to London, so does John. This may be due to the fact that Peter is working on a classified project and we want to hide the fact that John also works on the same project. By observing the pattern that Peter and John always travel together to London, one may infer the sensitive fact through association. The inference controller should detect the fact that a user may be able to infer this sensitive information and not release certain responses to the user.

Building an inference controller based on the second approach is extremely difficult as theory and foundations for data mining are yet to be developed. While there is some wok on the relationship between inductive logic programming and data mining, the research is still in the preliminary stages. Current data mining techniques are rather ad-hoc and therefore it is nearly impossible to build such an inference controller. Note that the work reported in [THUR95] takes a similar approach to handle the inference problem, but focuses only on deductive reasoning. Data mining techniques are far more complex than deductive reasoning.

The research reported in [CLIF96] shows much promise on developing techniques to handle the inference problem based on the first approach. For example, it has been shown that by applying various data mining tools that exist today, one could deduce some potentially sensitive information. The challenge then is to develop techniques to handle this problem. Some of the methods that are being explored include giving partial answers to queries, introducing additional information and noise into the responses, and giving answers to different but related queries. Research in this area is just beginning.

REFERENCES

[BLAU95] B. Blaustein, C. McCollum, A. Rosenthal, K. Smith, and L. Notargiacomo, "Autonomy and Confidentiality: Secure Federated Data Management," Second International Conference on Next Generation Information Technologies and Systems, Naharia, Israel, June 1995.

[CLIF96] C. Clifton, and D. Marks, "Security and Privacy Issues for Data Mining," Proceedings of the ACM SIGMOD Conference Workshop on Data Mining, Montreal, Canada, June 1996.

[GRIN96] G. Grinstein, "Data Exploration through Mining and Visualization," To be published in the Proceedings of the IEEE Visualization'96 Conference, San Francisco, CA October 1996.

[IEEE93] Special Issue on Data Mining, IEEE Transactions on Knowledge and Data Engineering, December 1993 (Ed: N. Cercone and M. Tsuchiya)

[INMO96] W. H. Inmon, "Building the Data Warehouse," John Wiley and Sons, 1993

[LIN95] T.Y. Lin, D. Marks, T. Hinke, and B. Thuraisingham, "Data Mining and Security," Special Session at the 9th IFIP 11.3 Database Security Workshop, N.Y. August 1995.

[MARK96] D. Marks, "Inference in MLS Databases," IEEE Transactions on Knowledge and Data Engineering, February 1996.

[THUR95] B. Thuraisingham and W. Ford, "Security Constraint Processing in a Multilevel Secure Distributed Database System," IEEE Transactions on Knowledge and Data Engineering, April 1995.

[ZHUG95] Y. Zhuge, H. Garcia-Molina, J. Hammer, and J. Widom, "View Maintenance in a Warehousing Environment," Proceedings of the ACM SIGMOD Conference, San Jose, CA, May 1995.

ACKNOWLEDGMENTS

I thank Don Marks of the Office of INFOSEC Computer Science, Department of Defense, and Cathy McCollum, Chris Clifton, Georges Grinstein, and Ken Smith all of The MITRE Corporation for comments and/or inputs to this abstract.

This paper was also published in the informal Proceedings of the 10th IFIP Working Conference in Database Security in July 1996. A detailed paper based on the IFIP presentation will be published in a book by Chapman and Hall in 1997. I thank Prof. Elisa Bertino, Prof. Pierangela Samarati, and Prof. Ravi Sandhu for the invitation to talk at the IFIP Conference. I also thank Dr. John Campbell of the DoD for his invitation to talk at the NISSC Conference.

DISCLAIMER

The views and conclusions reported in this paper are those of the author and do not reflect the policies and procedures of the MITRE Corporation or of the U.S. Government.

An Introduction to Data Warehousing, Data Mining and Security

Chair:
Dr. John R. Campbell, NSA

Panelists:
Bhavani Thuraisingham, The MITRE Corporation
Jesse C. Worthington, Informix Software Inc.

Data Warehousing is big and growing. Ken Rudin in an article in the August, 1996 Issue of DBMS, states that the Data Warehouse Market is currently $2B, and will be rising to $8B by 1998. What is a Data Warehouse? What is Data Mining? What is the history of data warehousing and mining? What are the problems in building a data warehouse? What are the benefits? What are the additional security considerations that should be considered when building a warehouse? What security considerations should be considered in data mining? This expert panel will take a brief look at these questions.

Panel

An Introduction to Data Warehousing, Data Mining and Security

Part 2: The Technology Issues

Chair:

John Davis,

Director

National Computer Security Center

Panelists:

Bhavani Thuraisingham, The MITRE Corporation

John Campbell, National Security Agency

Additional speakers will join us to discuss the isues which affect the long term security of Data Warehousing. This final session helps to set the stage for future Data Warehousing security solutions. On Friday, October 25, 1996, we will continue with the first of several workshops to be co-sponsored by the National Computer Security Center and the IEEE Mass Storage Committee.

Introduction to Information Warfare

Francis "Butch" Bondoc

Manager
Klein & Stump, Incorporated

bondoc@techplan.com

214 Washington Blvd., Suite 400
Arlington, Virginia 22204

Voice: (703) 415 - 9310
FAX: (703) 920 - 9626

ABSTRACT

This is a non-technical overview aimed at the newcomer to Information Warfare (IW).

We will introduce IW terminology, threats and countermeasures.

We will concentrate on Defensive Information Warfare and explore the solutions offered by the MISSI (Multilevel Information Systems Security Initiative) Architecture and the implementation of DMS (Defense Message System).

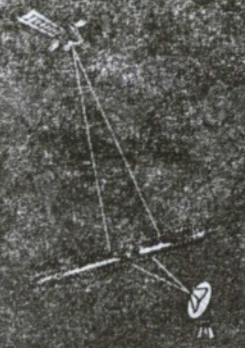
Information Warfare

INFOWAR

Klein & Stump Inc. - Butch Bondoc

Overview

- Definition
- INFOWAR & INFOSEC
- Defensive Information Warfare
- Tools, Approaches & Concepts
- Current Capabilities



Who has the Last Word?

INFOWAR Definition

- Martin Hill, OASD/C3I, IW definition: Actions to achieve information superiority by affecting adversary information, information-based processes and information systems, while defending one's own information, information-based processes and information systems.

INFOWAR & INFOSEC

- INFOSEC is one of the primary solutions
  - MISSI
  - DMS
- INFOWAR is the threat that requires it, and requires it

Defensive Information Warfare

- Pervasiveness and Accessibility will drive the defensive posture of INFOWAR
- Computerized information is accessible and at risk on a global basis
- If we can get to our information over open architectures, so can a hostile hacker

DIW *(cont.)*

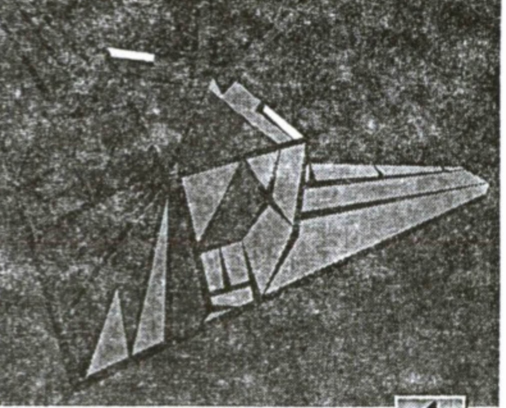
- Telecommunications Terrorists are out to do us and our information harm
- Security Perimeter around our information
  - No Gaps
  - No Loss of Functionality
- Through DIW we must protect the information that is the fabric of our lives

Beyond Tactical & Strategic

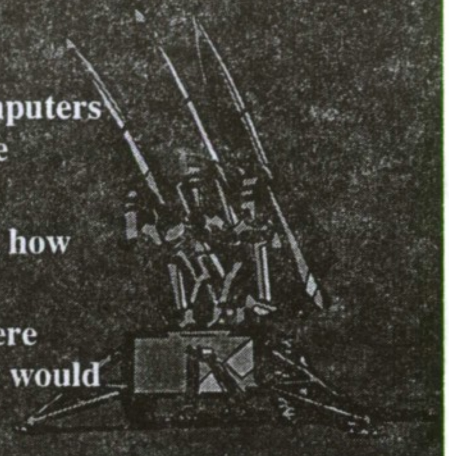
- Logistics, Finances
- Medical, Dental, Insurance
- Training, Travel, Transportation
- Communications and more ....
- All this information has been computerized and made more vulnerable to hostile forces

Specific Military Information-Based Processes

- GCI
- GCA
- IFF



C4IFTW

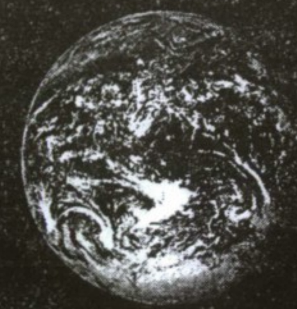


- Command, Control, Communications, Computers and Intelligence for the Warrior
- If the systems crashed, how would we go to war?
- If our own weapons were turned against us, how would we recover?

KSI - A TECHPLAN Company 8/1/96 9

Networks and the Internet

- LANs
- WANs
- Not physically & electronically confined
- Hook a PC to a modem and it leaps across the globe
- Reach out and touch someone



KSI - A TECHPLAN Company 8/1/96 10

Protection

- We need to protect the information itself
  - Label the information
  - Give the user an identity with privileges
- The old "need-to-know" in an electronic format
- Establish virtual secure enclaves amongst privileged users

KSI - A TECHPLAN Company 8/1/96 11

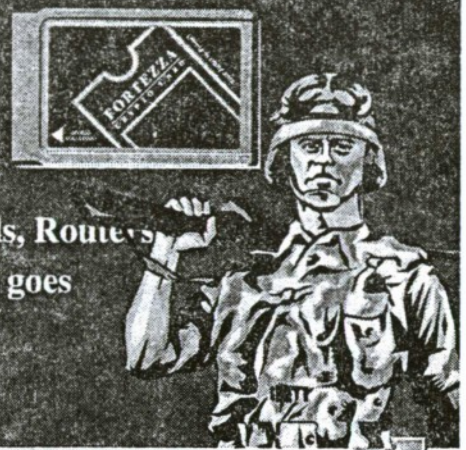
MISSI Security Services

- Integrity
- Identification & Authentication
- Non-Repudiation
- Confidentiality
- Availability

KSI - A TECHPLAN Company 8/1/96 12

Tools, Approaches & Concepts

- FORTEZZA Card
  - I.D.
  - Access Privileges
- CAW
- FIREWALLS, Guards, Routers
- When the Warfighter goes Tactical so must DIW



KSI - A TECHPLAN Company

8/1/96 13

Virtual Global Theater

- GCCS
- GCSS
- Warfighter no longer isolated in a local scenario
- The power of Information makes him aware of an expanded battlespace

KSI - A TECHPLAN Company

8/1/96 14

Desert Shield / Storm

- Multi-National forces supplied logistical and battle support over sea, air and land
- INFOSEC provided secure phones, secure radios, secure IFF, secure links and more ....

KSI - A TECHPLAN Company

8/1/96 15

SMI

- Security Management Infrastructure (SMI) providing common security foundation for MISSI products
- CAW
- Directory Server
- Mail List Agent
- Rekey Manager
- Audit Manager

KSI - A TECHPLAN Company

8/1/96 16

Multilevel Security Example

- HQ sends out an SBU message, everyone with a FORTEZZA card should be able to unscramble and read the message
- When a SECRET message is sent out, only the planners, mission leaders and the like can unscramble and read the message
- Hostile Hacker should only see a scrambled message

DMS

- The driving force behind the implementation of MISSI
- The replacement system both for AUTODIN and existing E-mail
- Mandatory system for sending secure organizational messages
- Critical to "owning" the electronic battlefield
- Cornerstone for C4IFTW

DMS *(cont.)*

- Strategic & Tactical
- Operates within digitized battlefield
- Uses existing network
- X.500 Directory Server
  - Supports other applications
  - Alleviates routing problems
- X.400 Protocol
  - Supports attachments, such as imagery
  - Alleviate need for separate transmission systems

Areas of Protection

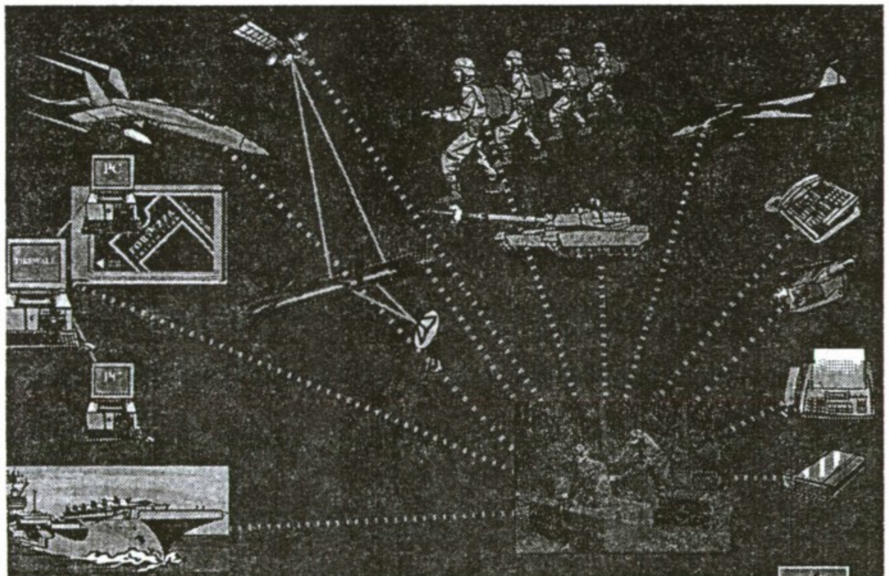
- Secure Voice
- Secure Datalink
- Secure Fax
- Secure Naval Broadcast
- Secure Radio
- Secure Satellite Comms
- Secure IFF
- Secure point-to-point
- Secure end-user to end-user

Current Capabilities

- STU-III
- STE
- VINSON
- VALLOR
- Other KG and KW Equipment
- The entire MISSI Suite

The MISSIng Piece

- All the pieces to inject security into the C4IFTW global scenario are here
- What's missing is a cohesive, robust security architecture to tie in media of various baud rates, synchronous modes, key generation and more into a single transmission medium to the digitized battlefield



INFORMATION WARFARE: REAL THREATS, DEFINITION CHANGES, AND SCIENCE FICTION

Chair: Wayne Madsen, Computer Sciences Corporation (CSC)

Panel Members:

Martin Hill, Deputy Director, Information Warfare (Programs), Office of the Assistant Secretary of Defense for C3I/Information Warfare
Frederick G. Tompkins/Matthew Devost: Science Applications International Corporation (SAIC)
Scott Shane: *The Baltimore Sun*
John Stanton: *Journal of Technology Transfer*

During 1996, the Information Warfare scenario has received a great deal of attention from national security planners, legislators, the military, intelligence agencies, the media (news and entertainment), and industry. The Department of Defense, the major focal point for IW, has altered some definitions within the IW arena. Other IW developments during 1996 will also be discussed, for example, some of the threats that experts view as being overly-hyped. Real and bon fide IW threats such as threats to C3I systems (electronic warfare, e.g.) will be separated from the science fiction realm of hand-held computer zap guns, HERF bazookas, and universal computer viruses. The ability (or inability) of governments to use technology to censor or otherwise manipulate information transmitted over networks, such as the Internet, will also be examined.

INFORMATION WARFARE: INDICATIONS AND WARNINGS

Compiled by: Wayne Madsen

Based on the assumption that most of the tactics surrounding Information Warfare consist of the control, surveillance, and manipulation of information by governments, the following indications and warnings (I&W) point to an increase in such practices by governments around the world.

Abkhazia In April 1996, Georgia and Russia agreed that all telecommunications links between the Republic of Abkhazia and other countries, including Russia, must pass through a telecommunications switch in Tblisi, the Georgian capital. The Georgian Ministry of Posts and Communications decided to halt unauthorized and unregistered voice and data communications between Abkhazia and other countries via Russia.

Asia-Pacific Region A majority of Hong Kong respondents to a poll asking them if indecent material on the Internet should be banned said no. There were also sizable opponents of such a ban in other countries in the region.

Respondents	*Percentage Against Internet Content Ban*
Hong Kong	52%
Indonesia	50
Western expatriates	48
Thailand	45
South Korea	41
Japan	39
Taiwan	38
Malaysia (including Sarawak and Sabah)	35
Australia	35
Singapore	31
Philippines	28
Asian expatriates	25

Source: *Far Eastern Economic Review*, July 11, 1996

Australia In August 1995, the Ministry for the Communications and the Arts directed the Australian Broadcasting Authority (ABA) to investigate content issues in the on-line information industry. The subsequent ABA "issues paper" suggested various methods to control Internet content, including blocking access to offensive sites."

New South Wales... The parliament is considering a law that would hold Internet Service Providers (ISPs) and individuals responsible for posting any sexually-explicit, drug-related, and crime-related information on the Internet.

Victoria Parliament passed a law which makes it an offense to use an on-line network to transmit "objectionable" material to minors.

Western Australia A law went into effect on January 1, 1996 that requires ISPs to censor "objectionable" and "restricted" materials to minors. Senders of such information are punished according to a very broad range of definitions.

Bahrain In December 1995, the government-owned telecommunications company, Batelco, provided on-line access to the Internet only after an expensive monitoring and filtering system was installed to block access to certain banned sites. *(Egyptian Gazette)*.

Canada On April 2, 1996, Justice Minister Allan Rock invited Canadians to present their views on regulating excessive violence in the media, including the Internet.

The Advisory Council on the Information Highway is formulating a policy on regulating content on the Internet.

A plan by Telesat Canada to finance its $1.6-billion satellite program by agreeing to lease some capacity to U.S. broadcasters resulted in the U.S. Federal Communications Commission deciding to hold special hearings to investigate whether it can regulate the use of Canadian satellites. (*Toronto Financial Post* 4 May 96).

China On June 2, 1994, two days prior to the fifth anniversary of the Tiananmien Massacre, the Public Security Ministry ordered major hotels in Beijing to suspend delivery of Cable News Network (CNN) broadcasts.

In late October 1995, China announced plans to change Taiwan's Internet domain name scheme (DNS) from `.tw` to `.tw.cn`.

According to an edict of the State Council issued on January 16, 1996, foreign news agencies were required to come under the centralized control of Xinhua, the official Chinese news agency. Information providers, including Reuters, Dow Jones-Telerite, the Associated Press, and Bloomberg, all of which sell economic news to China, were required to register with Xinhua within three months and domestic organizations were forbidden to buy economic information directly from foreign sources. Xinhua sets subscription rates for foreign vendors and those vendors providing information that slanders China are threatened with prosecution. Xinhua described the move as a means to protect state sovereignty and "protect the legal rights and interests of Chinese economic information users."

On February 1, 1996, new rules governing Internet access were issued by the government. All companies providing access are subject to official approval and all computer information networks are to use channels provided by the Ministry of Posts and Telecommunications (PTT) to link to networks abroad. Any existing networks were forced to disband and re-register. The PTT planned to install filtering software (firewalls) to prevent the reception of material from foreign sources known to offer pornography or "counter-revolutionary" ideas. The Chinese regulation stated that "no organization or individual may engage in activities at the expense of state security. Producing, retrieving, duplicating, or spreading information that may hinder public order are forbidden." On February 15, Xinhua announced a further decree under which all new users of international computer networks had to register with the security services within 30 days of linking to the Internet.

James Chu, the chief executive officer of China Internet, said "not just the Chinese government, but all governments, are concerned that information on the Internet could cause social instability."

Croatia During the height of the Yugoslav civil war, when censorship was imposed by all sides, Wim Kat, a professor in Zagreb, established a network of bulletin board systems called ZaMir. These systems were linked to an Internet server in Bielefeld, Germany. The embryonic network even extended to the besieged Bosnian city of Tuzla. Tuzla residents were able to pass uncensored information on Serbian genocide and other atrocities to computer users around the world.

Cuba Science, Technology, and Science Minister Rosa Elena Simeon said that Cuba must learn how to "use the Internet's capabilities and advantages while reducing its risks and disadvantages as much s possible." *(Inter Press Service).*

Cyberia Internet domain naming schemes (DNSs) are being used to establish political control over national network access to the Internet. Macedonia's `.mk` DNS was briefly suspended in 1994 after

Greece complained to the United States about implicit recognition of Macedonia by the use of `.mk`. China wants Taiwan's `.tw` DNS to come before `.cn`, a move that would imply Chinese control over Taiwan's Internet domain. East Timor's FRETILIN liberation movement and Western Sahara's POLISARIO movement have legitimate claims to use the DNSs `.tp` and `.eh`, respectively, since the United Nations has never recognized the two territories' forced annexations into Indonesia and Morocco. Moreover, the International Organization for Standards (ISO) still officially recognizes these territories as separate entities. There are no guarantees that certain uninhabited , nearly uninhabited, and extremely small islands possessing their own DNSs will ever be permitted to actively participate in the Internet, especially if they were "lent" to external interests such as anonymous remailers, environmental groups, or businesses. For example, a Norfolk Island entrepreneur is hoping to woo Australian companies to the Internet domain of `.nf`. Companies locating in Norfolk Island could evade the business registration requirements currently imposed in Australia. Other islands which could offer such "off-shore" Internet services are the Heard and McDonald Islands (`.hm`), Cocos (Keeling) Islands (`.cc`), British Indian Ocean Territory (`.io`), Svalbard and Jan Mayen (`.sj`), French Southern Territories (`.tf`), South Georgia and South Sandwich Islands (`.gs`), and Bouvet Island (`.bv`).

The following *Greetings from the Internet Liberation Front* was reported by *The Guardian*, July 12, 1995: "Once upon a time there was a wide area network called the Internet. A network unscathed by the capitalist Fortune 500 companies and the like. The somebody decided to deregulate the Internet and hand it over to the 'big boys' in the telecommunications industry . . . The Internet Liberation Front is a small, underground organization of computer security experts. We are capable of penetrating virtually any network linked to the Internet -- any network . . . Just a friendly warning Corporate America."

East Timor Jose Ramos-Horta, a representative of the National Council of Maubere Resistance, a coalition of Timorese political parties, said that Internet has become a primary tool for educating people inside and outside of Indonesia on the East Timor issue. East Timor was invaded and illegally occupied by Indonesia in 1975. Horta said "We have three Web sites, one in English, another in Portuguese, and another in the Indonesian language."

Egypt Although seven private Internet providers are now offering their services in Egypt a number of government, religious, and academic leaders are warning that the public should not be exposed to pornographic materials or subjected to an invasion of ideas that could threaten political stability and undermine Islamic culture. "If you have certain values you don't want them to be neglected," said the secretary-general of Egypt's Labor Party. He further said "Our society is Islamic, and we have our own values, which may not be the same as the West." (*Christian Science Monitor*, July 9. 1996

European Union On April 15, 1996, the European Union agreed to explore ways to regulate the Internet, Italian telecommunications minister Agostino Gambino said. "Many member states perceive the need now for some discipline, some kind of regulatory framework or code of ethics," Gambino said. Some EU governments, notably Germany and Great Britain, have already adopted Internet-related laws and others are considering it. France proposed at a meeting of EU telecommunications ministers in Bologna that countries draw up a global convention on ethical principles regarding the Internet and on regulations for the network. *(Wall Street Journal)*

The European Commission considered a directive dealing with interactive services such as the Internet. The European Parliament has introduced legislation that would amend the EU's broadcasting law to include such services as the Internet.

Finland In February 1995, the Finnish Criminal Police raided the residence of Johans Helsingius, an anonymous remailer provider in Helsinki, seizing computers and disks. The Finnish police were acting on a complaint from the Church of Scientology in Los Angeles (and the FBI's Interpol National Central Bureau, acting on behalf of the Scientologists). The religious sect claimed that copyrighted church materials were being posted illegally on a Usenet group called `alt.religion.scientology`. Finnish police discovered that one of the anonymous postings cam from an anti-Scientology activist in Britain whose identity was passed on by the police to the sect. The identities of other anonymous users were also discovered by the police. Shortly after the Finnish incident, U.S. Marshals raided the residence of another anti-Scientologist in Virginia, seizing computers and disks.

France Authors Jean-Marie Pontaut and Jerome Dupuis, in their book published in January 1996, *The Ears of the President*, alleged that telephone tapping is the rule rather than the exception in French politics. Telephone eavesdropping was particularly acute during the first seven years of President Francois Mitterand's term in office. The authors claimed that between 183 and 1986 the government illegally tapped the telephone lines of some 2,000 people, including 128 journalists. *(Reuters).*

In early 1996, the government was angered when, subsequent to its banning of *Le Grand Secret* by Claude Gubler, a book dealing with the health of the late President Mitterand, an electronic version of the book appeared on Internet websites abroad. The French Information Technology Minister called for the European Union to draft new legislation in order to regulate the Internet.

On June 7, 1996, the French Parliament (Senate and Assembly) passed an amendment to the French telecommunications regulation *law (Loi sur la Reglementation des Telecommunications (*LRT)). The law had been introduced by French telecommunications minister Francois Fillon. The law requires that ISPs must conform to future recommendations that will be establish by the *government's Comite Superieur de la Telematique (CST)* to regulate the content of text, images, and documents transmitted over the Internet. The CST was established in 1993 to regulate Minitel services (text and voice based services), by establishing a professional code of ethics. Under the new law, the CST will be responsible to the CSA (the French version of the FCC) which regulates radio and TV broadcasts. The ISPs will be required to block access to "blacklisted" Internet sites and newsgroups identified by the CST.

Germany In January 1996, CompuServe blacked out over 200 news groups and the news service Clarinet upon a court order by a Bavarian judge.

On January 26, 1996, Deutsche Telekom blocked subscribers from accessing alleged anti-Semitic websites on the T-Online Internet service. Three U.S. universities immediately mirrored some of the material on their own web sites in protest of the ban. This made the Deutsche Telekom move largely irrelevant.

In January 1996 Deutsche Telekom blocked access to the Santa Cruz, California-based Web Communications because it provided access to a neo-Nazi site in Canada. Web Communications said that while it did not agree with the material contained in the site, it was not the company's policy to censor its users. *(Reuters and San Jose Mercury News).*

In February 1996, Bundestag President Rita Suessmuth told the German parliament that "freedom of expression reaches its limit when human dignity is violated and violence is promoted" by the information superhighway.

In late March 1996, the German government said it would introduce Internet censorship legislation in mid-1996. The legislation would punish ISPs only if they knowingly permit "illegal material" on their services but would not expect ISPs to be responsible for all the content on their servers. *(Reuters).*

The Internet expert for the Social Democratic Party severely criticized the government's policies to censor the Internet. The criticisms were published in press releases and postings in Usenet news groups.

Guyana The government has announced that before it permits a full Internet gateway to be installed, it will require all Internet communications to be monitored and that it "would move to prevent any unauthorized installation of Internet services." The government (also said it was studying ways of regulating links to the Internet. *Latin American Weekly Report).*

Hong Kong On June 17, 1994, China attacked Hong Kong's plans to institute freedom of information provisions for the public's access to certain official information. Xinhua, the Chinese news agency, said the proposal "violated provisions of the Sino-British Joint Declaration."

On March 3, 1995, seven Internet service providers (ISPs) were raided by the Commercial Crime Bureau of the Hong Kong police. Computer equipment and data files were seized. At first, the ISPs were charged with aiding and abetting computer hackers. When that charge was widely ridiculed, a second charge was brought. The ISPs were charged with operating without a mandatory Public Non-Exclusive Telecommunications Service (PNETS) license issued by the Hong Kong Office of the Telecommunications Authority (OFTA). In reality, it is believed that the police could have been acting on behest of the Beijing authorities who were anxious to test the feasibility of shutting down Hong Kong's Internet connections after they assume control of the colony in 1997. The police action adversely affected some 10,000 users (some 60 per cent of Hong Kong's Internet users). Some of the Internet users affected rely on Internet for their livelihood.

In January 1996, the secretary of Hong Kong's Recreation and Culture Branch which regulates the on-line media, said that developments in other countries, particularly the United States, would be taken into account in formulating Hong Kong's policy. *(South China Morning Post).*

In March 1996, Peter Cheung Po-tak, the Commissioner of the Television and Entertainment Licensing Authority, said that Internet controls should ensure a "minimum degree of decency" to protect children. *(South China Morning Post).*

A 1996 report titled *The Internet in Asia* written by the Political and Economic Risk Consultancy (PERC) stated that the development of the Internet in Hong Kong could serve as a signpost of how the transition to Chinese rule would shape the business environment of the territory. The report pointed to "real fears that China might try to clamp down on the Internet." The report continued by stating that the Internet could be one of the "first battlefronts determining where Hong Kong's authority ends and China's authority begins in matters affecting both places." *(The Canberra Times).*

India The Department of Telecommunications (DOT) requires that ISPs regulate the transmission of objectionable and obscene material over the Internet. In January 1996, the DOT required all ISPs to route their communications through the state-owned VSNL phone company enabling the government to monitor the Internet more effectively.

Indonesia In 1995, the government announced plans to train computer operators to post data on the Internet that would counter "bad information"` about the country. The operators would also be trained on how to gather military intelligence on-line. Indonesia's government is particularly upset about George Aditjondro, an Indonesian exiled in Australia who posts articles critical of President Suharto's family's business dealings and the policy of the government in East Timor. An Indonesian army general charged that Aditjondro was using Internet to promote "communist" agitation. *(Sunday Telegraph (London).*

Information Minister Harmoko suggested on December 6, 1995, that the government's main concern with the Internet is politics rather than pornography. He said that his ministry would monitor the Internet for "matters harmful to national security."

In November 1995, Armed Forces spokesman Brig. Gen. Surwarno Adiwijoyo suggested that the Communications Ministry might have to institute a "toll gate" in order to black out objectionable news that could damage Indonesian culture or adversely affect national security. He also indicated it may be necessary to register Internet users and ban access to certain news groups. *(Human Rights Watch).*

Iran On April 21, 1995, the government banned private television satellite dishes. Certain police units were authorized to raid homes in order to remove dishes. *(Freedom House).*

In August 1995, the telecommunications link of a private ISP were severed by the government-owned Telecommunications Company of Iran after reports that young people were using the service for objectionable conversations. *(Middle East Economic Digest).*

Italy On May 10, 1994, Italian police raided the locations of several Fidonet users. Although police said they were interested in cracking down on hackers who had allegedly illegally copied proprietary software and obtained passwords, one of the bulletin boards shit down was "BITS Against the Empire," a popular anarchist board that contained Trotskyite and other left-wing information.

Jordan Jordan contracted with the U.S. firm GlobeNet to provide Internet access with a firewall to allow Jordanian censors to preview material before it is transmitted to Jordanian subscribers. Carlton Tolsdorf, vice president of GlobeNet said, "We agreed with the authorities' request. And, by the way, I think we should have the same thing back home in the United States." (Scripps Howard article, "Arab World grapples with the Internet's benefits," January 7, 1996).

Kenya In 1995, Kenya belatedly became the twelfth African country to gain full Internet access. Kenya's reluctance stemmed from the fears of Kenyan President Daniel arap Moi that Kenyans will be influenced by uncensored democratic ideas from the rest of the world.

Kuwait The Communications Ministry said in April 1996 that a new Internet service planned for the country could only be made operational after the ISP ensured that "no pornography or politically-subversive" material is made available in Kuwait. *(Agence France Presse).*

Malaysia In September 1995, Information Minister Datuk Mohamed Rahmat condemned dissent on the Internet. He said that Malaysian students abroad were smearing the good name of Malaysia and that the government was considering laws to curb such "abuses." Rahmat suggested that those seeking information concerning Malaysia read the on-line editions of officially-sanctioned newspapers such as *The Star* and *Berita Harian.*

Anwar Ibrahim, the Deputy Prime Minister (and presumptive Prime Minister), warned against Internet censorship. "Let us not forget, that an informed citizenry is also a responsible citizenry," he said. On March 7, 1996, Ibrahim said while speaking at the Internet World '96 conference on March 7, 1996 that the government had no plans to censor the Internet. Ibrahim said that "Simply closing our doors will not only hurt us but will push us back in the race for growth and prosperity."

On April 3, 1996, Prime Minister Mahathir Mohamed spoke of the need for international action to stop "dirty literature from flowing to other nations" over the Internet.

In March 1996, Information Minister Rahmat announced plans to set up a new body to regulate the Internet. He said those criticizing the government "will face the music. " *(South China Morning Post).*

Mexico During the armed Indian rebellion in Chiapas in 1994, Commandante Marcos, the masked leader of the Zapatista National Libation Front (EZLN), used Internet to transmit communiqués to supporters and media around the world. Marcos used a laptop computer connected to a cellular telephone which was powered by an AC adapter plugged into his Jeep's cigarette lighter. Efforts by the Mexican military to pinpoint Marcos's location by conducting radio direction finding were unsuccessful.

Morocco... The state post and telecommunications company, ONPT, introduced Internet service on November 16, 1995. Commercial companies providing Internet service are required to comply with all government regulations regarding the operation of the service.

Mozambique After becoming one of the latest countries to link to the Internet, President Joaquim Chissano issued a decree creating a new Information Office, a component of the Prime Minister's Office.

Myanmar The Free Burma coalition home page (http://danenet.wicip.org/freeburma.html) has been successful in organizing an international boycott of companies that do business with Burma (Myanmar). The home page offers speeches by Burmese pro-democracy leader Aung San Suu Kyi. Whenever the Burmese junta commits atrocities or other abuses against citizens, information is carried out of Burma on diskettes. The information is immediately posted to the Internet. Another electronic mailing list, BurmaNet, has thousands of subscribers in 15 countries and posts news stories obtained from Bangkok newspapers and communiqués by ethnic rebel groups that are smuggles out of Burma on diskettes. Many rebel groups are armed with laptop computers and use Pretty Good Privacy (PGP) to encrypt the data on their diskettes and hard drives.

In May 1996, James Leander Nichols, an Anglo-Burmese businessman and the Honorary Counsel for Switzerland, Finland, Norway, and Denmark in Burma and a close associate of pro-democracy leader Aung San Suu Kyi, was sentenced to three years in prison for having two fax machines and a telephone switchboard with nine lines in his home. In order to discourage contact between Burmese citizens and the outside world, Burma's military regime, known as the "SLORC" requires Burmese to get the government's permission to own a fax machine, satellite dish, or sophisticated phone system. *(The Atlanta Constitution/The Atlanta Journal)*. Nichols later died in prison under mysterious circumstances.

New Zealand The New Zealand Technology and Crimes Reform Bill would sever all users from any site that was found to have transmitted a single piece of "objectionable" material to a single user. Under current law, the police may shut down any ISP found to contain any material deemed objectionable. "Objectionable" is defined as any information dealing with sex, horror, crime, cruelty, or violence that is likely to be injurious to the public good.

Norway In February 1996, Prime Minister Gro Harlem Brundtland stated that it is not possible to regulate the flow of information on the Internet. She said national censorship cannot regulate the rapid changing world of information technology. Commenting on the U.S. Communications Decency Act, Brundtland said that "They [the Americans] won't be able to regulate the Internet, it cannot be controlled." The Norwegian press hailed Brundtland's remarks. One paper editorialized that "control of information has, through the years, been the key to oppression and control of peoples. But to the vines of this invisible beanstalk, which are permeating every layer of society, every facet of business life, every corner of the globe, the legislative bodies of the world seem to be at a loss, helpless, and redundant." (*Nordiske Tidende*, February 29, 1996).

Pakistan A spokesman for the National Institute of Electronics stated in 1995 that Pakistan would limit Internet access to a small number of nodes and hosts. The government, he noted, would require ISPs to monitor and interdict undesirable discussion groups and electronic mail. *(Reuters)*.

Peru The Peruvian government has been waging a war with the Shining Path *(Sendero Luminoso)* guerrilla movement in cyberspace. Peru and the Shining Path have attempted to identify the locations of each other's computers in Peru and other countries in order to erase all the information contained by them. Ironically, Peru and the Shining Path have been using the same Internet server in New York.

Philippines... In March 1996, various Internet censorship bills were introduced by the Philippines legislature.

Republic of Korea In 1995, the Information and Communications Ethics Committee of the Data and Communications Ministry said local computer networks would be asked to prohibit access by South Koreans to sites containing sexually explicit or undesirable material such as information deemed "subversive." The subversive category includes information on bomb making and drugs.

On June 6, 1996, South Korean prosecutors said stern measures would be taken against any South Korean who attempted to read North Korean home pages on the World Wide Web. The prosecutors cited the National Security Law barring all unauthorized contact with the North. South Koreans distributing or downloading North Korean information would be punished. Although North Korea possessed no direct links to the Internet, some home pages abroad carry North Korean news and information. It is also against the law to possess any North Korean propaganda.

Russia The Russian Foreign Intelligence Service (SVR) has charged that the Soros Foundation, which funds the Open Media Research Institute (OMRI) in Prague (the former Radio Free Europe/Radio Liberty), is a CIA front. OMRI's daily digest of events in eastern Europe and the Commonwealth of Independent States attracts over 11,000 Internet subscribers and I among the six most popular services on the Internet.

On April 3, 1995, President Boris Yeltsin signed a decree that empowered the Federal Agency of Government Communications and Information (FAPSI) to approve all encoding devices used by the government, government enterprises, and banks. Russian companies providing encryption services and devices must be licensed by FAPSI. Foreign devices are prohibited from the country without a license from the Ministry of Foreign Economic Relations, issued in cooperation with FAPSI.

Saudi Arabia Dr. Ali al-Johani, the Minister for Posts, Telephones, and Telegraphs said in February 1996 that, although the Internet was difficult for the government to control, Saudi authorities were investigating ways to regulate it. *(Arab News)*.

On April 8, 1996, the Saudi government halted Saudi Orbit satellite broadcasts of the British Broadcasting Corporation's Arabic television service. In January 1996, the Orbit satellite relay station in Rome began to selectively black out portions of BBC news broadcasts which the Saudis found to be politically objectionable.

Singapore In 1994, Singapore's government scanned 80,000 Internet files and issued warnings to users of five files found to contain "pornographic" material. This resulted in concern that Singapore's plans to develop the country into an "information hub" for Asia would result in sensitive commercial information being intercepted by Singapore authorities. (*The Australian*, June 28, 1996)

In March 1996, Singapore introduced "anti-pollution measures" to clean up the Internet in Singapore. The three sole local providers offering access to the Internet are required to filter out offensive material including information that does not conform to "local values." Cyber cafes, schools and libraries must install filtering software such as NetNanny and SurfWatch and are held responsible for censoring the content of users and supervising the use of public Internet terminals. Also, political and religious organizations posting information on the World Wide Web are required to register with the government's Singapore Broadcasting Authority. Foreign on-line newspapers seeking Singapore subscribers are required to register with the government and comply with the same restrictions applied local newspapers. The government currently blocks access to more than half of the Usenet newsgroups available on the Internet.

In July 1996, the Singapore Broadcast Authority announced guidelines to control political, religious, and pornographic content on the Internet. The authority required all Internet operators to register with the government after July 15, 1996.

Loyalists of the governing People's Action Party (PAP) routinely scour the Internet in order to battle against "misinformation" posted about Singapore on the net. The PAP "cyber-battalions" are particularly interested in the newsgroup soc.culture.singapore which often contains information derogatory to the Singapore government. Referring to such political discussion groups on the Internet, Senior Minister Lee Kuan Yew pontificated that only the "top 3 to 5 percent of a society can handle this free-for-all, this clash of ideas." (*The Australian*, June 28, 1996).

The Singapore Broadcasting Authority (SBA) said that although political parties will need government licenses, it was not clear if individual politicians would be allowed to post anti-government views on bulletin boards. The SBA guidelines state that it will not permit contents that "tend to bring the government into hatred or contempt, or which excite disaffection against the government." The definition of hatred or contempt has not been determined. The government will also ban: "contents that jeopardize public security or national defense," "anything that ridicules racial or religious groups." "the promotion of religious deviations or occult practices." "the 'gross exploitation' of violence, nudity, sex or horror," and "the depiction of 'sexual perversions' such as homosexuality." *(Independent Television News).*

Slovakia On March 26, 1996, the Slovak parliament passed an amendment to the Penal Code prohibiting the dissemination of false information concerning Slovakia abroad, including information transmitted electronically. The law was attacked by opposition political leaders as excessively broad in scope and Catholic bishops condemned the law as morally reprehensible. On April 4, 1996, President Michal Kovac refused to sign the law and returned it to parliament.

South Africa On October 19, 1995, one of South Africa's most popular hard copy and on-line newspapers, the *South African Times*, was denounced as racist by Gauteng Premier Tokyo Sexwale.

President Nelson Mandela, delivering the keynote address to the Telecom '95 conference in Geneva said, "The information revolution cannot be rolled back."

Thailand... In February 1996, the National Electronics and Computer Technology Center (NECTEC) announced that it was requiring Internet subscribers and service providers to agree not to post anything the government considered to be indecent or they would be prosecuted.

United Arab Emirates In early 1996 the police held a seminar on restricting political use of the Internet as well as combating pornographic material. *(Reuters)*

Abu Dhabi The local Internet club in the emirate has agreed to ban the discussion of sex, religion, and politics on the Internet in order to respect local laws. *(Reuters).*

United Kingdom In March 1996, Trade and Industry Minister Ian Taylor urged ISPs to adopt a voluntary code of practice relating to Internet content. Taylor expected the code to cover "both illegal and undesirable material." Taylor warned that in the absence of such a voluntary code there would be "increased political pressure for legislation in various areas." *New Media Age (London).*

On April 16, 1996, the High Court granted the Department of Trade and Industry and injunction banning *The Economist* from publishing any further details from a leaked report by the Monopolies and Mergers Commission. The information concerned a merger in the electrical utility industry.

United States As of mid-June 1995, America On-Line (AOL) was reporting cutting off six users a day for "net abuse." In December 1995, AOL banned the use of the word "breast" -- effectively shutting down a users' group dealing with breast cancer. Prodigy and CompuServe were also reported to be conducting increased monitoring and banning certain content.

On February 8, 1996, President Bill Clinton signed the omnibus Telecommunications Act which made it a criminal offense to knowingly put "indecent" material on the Internet so that it could be viewed by a minor. Later in February, a federal judge in Philadelphia stayed the implementation of the law as a result of a lawsuit brought by civil liberties groups and on-line providers. Senators Leahy and Feingold introduced legislation to repeal the measure while House Speaker Newt Gingrich had earlier questioned the law's constitutionality.

On June 12, 996, the three-member appellate panel of the U.S. District Court for the Eastern District of Pennsylvania unanimously ruled that the Communications Decency Act was unconstitutional. In his opinion, Judge Stewart Dalzell stated "As the most participatory form of mass speech yet developed, the Internet deserves the highest protection from government intrusion . . . Just as the strength of the Internet is chaos, so the strengths of our liberty depends upon the chaos and cacophony of the unfettered speech the First Amendment protects." The Clinton administration quickly announced plans to appeal the judicial decision to the Supreme Court.

On April 4, 1996, the World Wide Web consortium announced plans to introduce a new communications protocol called the Platform for Internet Content Selection (PICS) that parents could use to monitor what information their children access on the Internet. The protocol was developed amid fears that the U.S. government was planning to censor Internet sites. On March 8, 1996, President Clinton announced that the chiefs of the U.S. television industry had agreed to a rating system to be used in concert with the "V-chip" censorship technology. The agreement had been reached after tremendous pressure had been brought on the industry by the administration.

Parents, not governments, ultimately are and should be responsible for what their children watch. Technology is starting to provide them with more and more sophisticated means of doing so. Channel blocking devices are already widely available. Soon, parents will be able to avail themselves of software that blocks programming case by case.

Or they jut turn the thing off. That was not an option available to Winston Smith in George Orwell's "1984." The set droned on and on, purged of programming the government found "objectionable," and Smith took whatever comfort he could find from his ration of government-produced Victory Gin and Victory Cigarettes.

We don't need a Victory Chip.

The Washington Times, Editorial, July 12, 1995

In June 1996, CIA director John Deutch said the CIA was working with the FBI and Justice Department to collect information about computer hackers and their activities. Deutch said the information was being collected from both informants and from other advanced means, including signals intelligence. *(Defense Daily*, 26 June 1996)

In June 1996, CIA director Deutch told the Senate Government Affairs Committee that he had ordered a National Intelligence Estimate (NIE) to be conducted on information warfare, Deutch said that even the smallest radical group can exploit the unregulated and undefended expanse of cyberspace. He cited the Islamist radical group Hezbollah as being one group that has successfully used the Internet and other advanced communications technologies for their daily operations. The NIE, due to be completed by December 1, 1996 will also include comments from the U.S. law enforcement community, the Defense Information Systems Agency, the Departments of the Army, Navy, and Air Force, the National Security Agency, and the major telecommunications providers.

Vatican City French Bishop Jacques Gaillot, ousted from his Evreux See for his liberal views, established the first "virtual diocese" in cyberspace. The site (http://www.partenia.fr) was jammed by thousands of Internet users when it came on line. The bishop's Internet site has annoyed the Vatican because the bishop's views conflict with the Holy See's positions on HIV-AIDS and contraception.

Vietnam In January 1996, Pham Dao, director of the state-owned Vietnam Datacommunications Company (VDC), confirmed that it will censor the Internet connection to Vietnam in order to comply with government regulations. He said an Internet firewall would be installed which would screen out transmissions from specific senders and news sources. Another VDC official stated Vietnam's desire to control the Internet "is the requirement from our leaders, our government. The Internet must be controlled, not only for technical and security reasons but from the cultural aspect."

The government has decided that the Ministry of Culture and Information will be responsible for monitoring on-line contact via the Internet and the Interior Ministry will be charged with monitoring Internet national security issues. The government has sought assistance from Singapore in policing the Internet in Vietnam. *(British Broadcasting Corporation).*

Yugoslavia A Croatian spy ring was arrested in Yugoslavia for smuggling Yugoslav military secrets out of the country on computer game diskettes. The masked information included information on missile units, combat readiness of aircraft, and early warning and missile guidance systems. The military information text files were disguised as game titles.

Zambia The February 5, 1996 Internet edition of *The Post*, one of the country's most important opposition newspapers, was banned by President Frederick Chiluba under the State Security Act. Although the paper remained on the web for two days after the banning, the police warned the ISP, Zamnet Communications, that it would be criminally liable if it did not remove the electronic edition in question.

Panel

Security in World Wide Web Browsers: More than Visa cards?

Moderator
Rob Dobry
Workstation Security Products Division
National Security Agency
(410) 859-4464

Panelists

Representatives from Vendors working with Web Browsers and Web Security

Abstract:

Currently, security in world wide web browsers is frequently discussed in the context of being able to handle electronic commerce via the net. LL Bean wants to sell boots 24 hrs a day via the Web, and consumers want some confidence that little Joanna Hacker isn't buying printers and modems with the same credit card. Various DOD programs would like to use web browsers as a system independent interface to information systems. Java may be a help, but are the security protocols there (or coming), and will the flexibility support other security problems?

Panel

Attack/Defense

Moderator
Jon R. David
The Fortress
P.O. 731
New City, NY 10956-0731
(914) 365 - 4700
fortress@dockmaster.ncsc.mil

Panelists

Steve Bellovin
AT&T Bell Laboratories

Murray Hill, NJ 07974
smb@research.att.com

Padgett Peterson
Lockheed-Martin
MS423 12506 Lake Underhill Rd.
Orlando, FL 32825
padgett@tccslr.dnet.mmc.com

Bill Cheswick
Bell Laboratories, Lucent Technologies
Room 2C-416, 700 Mountain Ave.
Murray Hill, NJ 07974
ches@bell-labs.com

Marcus Ranum
V-ONE Corporation
1803 Research Blvd.
Rockville, MD 20850
mjr@v-one.com

DESCRIPTION:

The Internet has proven to be so vulnerable that the goals of security practitioners have changed. Keeping the bad guys out is no longer the prime goal of security, rather the prompt and accurate identification of intrusions (or, preferably, intrusion attempts) and minimizing the damages. While foolproof security is not a realistic goal for any computer system, it is therefore less likely to be realized with Internet involvement.

Media exposure, public advisories (CERT, ASSIST, CIAC, etc.), vendor bulletins, Internet forums and the like have detailed quite a few "popular" Internet attack methodologies. While these are certainly not the only dangers from which users need protection, the publicity they've gotten make them the most likely attacks. This session examines these "popular" attacks and presents ways to effectively defend against them.

Session 1
The Web -- What is it, Why/How is it Vulnerable?

Session 2
Securing the Web

Moderator & speaker:
Jon R. David
The Fortress
P.O. 731
New City, NY 10956-0731
(914) 365 - 4700
fortress@dockmaster.ncsc.mil

Panelists:

Jon Freivald
Charter Systems, Inc.
461 5TH Avenue, 15TH Floor
New York, New York 10017
(212) 447 - 0550 (Voice)
(212) 447 - 0552 (FAX)

Padgett Peterson
Lockheed-Martin
MS423 12506 Lake Underhill Rd.
Orlando, FL 32825
padgett@tceslr.dnet.mmc.com

Drew Dean
Department of Computer Science
Princeton University

DESCRIPTIONS:

While many users view "the Internet" and "the web" as one in the same thing, they are totally different -- in operation, in appeal, and in exposures. In spite of the great impact that the Internet has had on our business and personal lives, the web seems to be catching the user fancy at a significantly higher rate. Web security is therefore critically important, and will be addressed in a 2-part approach:

The first session, "The Web -- What is it, Why/How is it Vulnerable," will formally describe what the web is/does, indicate how it differs from "normal" Internet use, show it in typical/popular operational modes, and point out the nature and magnitude of primary vulnerabilities.

The second session, "Securing the Web," will show how to treat the vulnerabilities uncovered in the first session in and of themselves, and as a part of both Internet security programs and total security programs.

Electronic Data: Privacy, Security, Confidentiality Issues

Moderator:
Kristin R. Blair, Esq., Duvall, Harringan, Hale and Hassan

Panelists:
The Honorable Leslie M. Alden, Fairfax County Circuit Court
Steve A. Mandell, Esq., The Mandall Law Firm
Ronald J. Palenski, Esq., Gordon and Glickson P.C.
Steven W. Ray, Esq., Kruchko and Fries

The rapid expansion in the use of electronic data and computerization in the past decade has led to a rapid expansion in the legal arena. While the burgeoning use of electronic data increases profits and efficiency, it creates potential pitfalls for businesses that must be recognized and avoided.

In the employment area, there are competing interests between employer and employee due to the capability of employers to monitor most of the activities of employees on work premises. In certain circumstances employers may videotape their employees. In other circumstances employers may monitor employee telephone calls, email and voice mail. Within legal limits employers may subject their employees to polygraph examinations. It is essential that employers disseminate policies to their employees regarding their electronic monitoring practices and procedures in order to avoid legal exposure.

Employees are striving through legal means to attempt to preserve their decreasing privacy. Lawsuits brought by workers against their employers will continue to increase in number and complexity when employees feel that their privacy rights have been violated. Employers need to be regularly updated on developments in employment law. Some companies have been able to adequately balance the interests of employees and employers regarding the privacy issues, and all companies should be striving to reach this balance.

Businesses pour huge amounts of money into computers and the information stored within them and need to be able to protect their investments from thieves, vandals and competitors. Businesses must protect their computer systems from a wide barrage of threats including hackers, disgruntled employees, viruses, worms and other damaging programs. Computer law is a rapidly growing body of law, and new laws pertaining to computers are being written and passed. For example, Virginia passed the Virginia Computer Crimes Act because many computer crimes simply were not addressed by the common law. On-line obscenity laws are developing as the government seeks to regulate indecent material which computer users attempt to disseminate for pleasure or profit.

Intellectual property law continues to change as case law determines the appropriate protection for computer programs and other data. Recent case law shows battles over on-line trademark infringement through the improper use of domain names.

In summation, the use of electronic data poses both new opportunities to enhance the interests of business through monitoring employees and protecting the work product as well as new potential areas of corporate liability. This fascinating area of the law will continue to grow and change in the coming years.

VIRGINIA COMPUTER CRIME LAW
The Honorable Leslie M. Alden

I. THE COMMON LAW DID NOT ADDRESS COMPUTER CRIME.

Established criminal laws were not equipped to deal with the new specie of crime generated regarding the use of computers. The often intangible nature of the computer products or services simply did not come with definitions established by existing legislation and case law.

A. Lund v. Commonwealth, 217 Va. 688 (1977)

The unauthorized use of computer time and services could not form the basis of a larceny conviction as these were not "goods and chattels."

B. Va. Code Section 18.2-98.1 was enacted in 1978 to provide that computer related services and data constituted property which may be the subject of larceny, embezzlement and related property crimes.

II. THE ACT, SEEMINGLY, FILLS SOME OF THE GAPS.

The Virginia Computer Crimes Act (the "Act") was first enacted in 1984 and created both civil and criminal penalties for victims of computer related conduct. The Act was intended primarily to meet deficiencies in the common law criminal arena regarding acts related to computers by creating new offenses, rather than by straining to fit the prohibited acts within the existing criminal law framework.

A. The Act begins by defining several words and phrases, including the terms: computer, computer data, computer network, computer operation, computer program, computer services, computer software, financial instrument, owner, person, property, uses, and without authority.

B. The Act defines five new crimes:

1. Computer Fraud -- 18.2-152.3
Value over $200 is Class 5 felony

2. Computer Trespass -- 18.2-152.4
If done maliciously and value of damaged property exceeds $2500, Class 6 felony.

3. Computer Invasion of Privacy -- 18.2-152.5
   Class 3 misdemeanor

4. Theft of Computer Services --
   18.2-152.6

5. Personal Trespass by Computer --
   18.2-152.7
   If malicious, Class 3 felony

C. The prosecution of a misdemeanor under the Act must commence within the earlier of five years after the commission of the last act, or one year after the existence of the prohibited act is discovered. 18.2-152.9

D. Venue provisions have been expanded and set out. 18.2-152.10

E. Section 18.2-152.14 provides that the absence of a tangible document created or altered by the offender shall not be a defense to a charge of forgery under traditional common law principles.

F. Section 18.2-152.12 provides for civil relief, including recovery for any damages sustained, including lost profits. Also provides that legal proceedings may be conducted in any way necessary to protect the trade secrets and security of the owner.

III. THE CASE LAW HAS NOT EXTENDED THE PROTECTIONS OF THE LAW VERY FAR

A. Rosciszewski v. Arete Associates, Inc., 1 F.3d 225 (4th Cir. 1993). Plaintiff claimed that defendant accessed plaintiff's computer system and procured copies of a copyrighted computer program and other proprietary works. The court held that the protection of computer programs from unauthorized copying granted under 18.2-152.3 is equivalent to the exclusive right of the copyright owner to reproduce a copyrighted work under the Copyright Act. The core of both a cause of action under section 301(a) of the Copyright Act and section 18.2-152.3 of the computer crime law, in the context of this case, is the unauthorized copying of a computer program. Accordingly, because this claim of a violation of 18.2-152.3 does not require proof of elements beyond those necessary to prove copyright infringement of a computer program, the Copyright Act completely preempts such a claim under Virginia state law.

B. O'Connor v. Commonwealth, 16 Va. App. 416 (1993). This larceny/false pretense conviction was overturned because the set of specifications received by the defendant was neither "computer software" nor a "computer program" as defined by 18.2-152.2.

IV. OTHER COMPUTER RELATED STATUTES WHICH MAY PROVIDE PROTECTION

A. Computer Abuse Amendments Act of 1994 -- codified in 18 U.S.C. Section 1030 (1994) as amendments to the Computer Fraud and Abuse Act. This legislation, included as part of the crime bill signed by President Clinton, prohibits the unauthorized transmission in interstate commerce of code that is intended to damage or deny use of a computer system if the damages are more than $1000 during any one year period.

B. Proposed U.C.C. Self-Help Remedy. The National Conference of Commissioners on Uniform State Laws is considering an expansion of Article 2 of the U.C.C. expressly to cover software transactions. The proposals deal primarily with the ability of a licensor to repossess computer related intangibles in the event of a material breach of the contract of lease or sale.

C. Virginia's Attempt to Create Additional Criminal Penalties. In 1994, the General Assembly passed legislation that would have prohibited vendors from embedding disabling devices in software programs. After objections by the industry, Governor Allen required the legislation to be passed by two consecutive sessions of the assembly. The legislature next attempted to require vendors to provide written notice that a software package contained a disablement code or be guilty of a Class one misdemeanor. None of this legislation yet has been adopted.

APPENDIX

Article 7.1.
Computer Crimes.

§ 18.2-152.1. Short title.

This article shall be known and may be cited as the "Virginia Computer Crimes Act."

§ 18.2-152.2. Definitions.

For purposes of this article:

"Computer" means an electronic, magnetic, optical, hydraulic or organic device or group of devices which, pursuant to a computer program, to human instruction, or to permanent instructions contained in the device or group of devices, can automatically perform computer operations with or on computer data and can communicate the results to another computer or to a person. The term "computer" includes any connected or directly related device, equipment, or facility which enables the computer to store, retrieve or communicate computer programs, computer data or the results of computer operations to or from a person, another computer or another device.

"Computer data" means any representation of information, knowledge, facts, concepts, or instructions which is being prepared or has been prepared and is intended to be processed, is being processed, or has been processed in a computer or computer network. "Computer data" may be in any form, whether readable only by a computer or only by a human or by either, including, but not limited to, computer printouts, magnetic storage media, punched cards, or stored internally in the memory of the computer.

"Computer network" means a set of related, remotely connected devices and any communications facilities including more than one computer with the capability to transmit data among them through the communications facilities.

"Computer operation" means arithmetic, logical, monitoring, storage or retrieval functions and any combination thereof, and includes, but is not limited to, communication with, storage of data to, or retrieval of data from any device or human hand manipulation of electronic or magnetic impulses. A "computer operation" for a particular computer may also be any function for which that computer was generally designed.

"Computer program" means an ordered set of data representing coded instructions or statements that, when executed by a computer, causes the computer to perform one or more computer operations.

"Computer services" includes computer time or services or data processing services or information or data stored in connection therewith.

"Computer software" means a set of computer programs, procedures and associated documentation concerned with computer data or with the operation of a computer, computer program, or computer network.

"Financial instrument" includes, but is not limited to, any check, draft, warrant, money order, note, certificate of deposit, letter of credit, bill of exchange, credit or debit card, transaction authorization mechanism, marketable security, or any computerized representation thereof.

"Owner" means an owner or lessee of a computer or a computer network or an owner, lessee, or licensee of computer data, computer programs, or computer software.

"Person" shall include any individual, partnership, association, corporation or joint venture.

"Property" shall include:

1. Real property;

2. Computers and computer networks;

3. Financial instruments, computer data, computer programs, computer software and all other personal property regardless of whether they are:

a. Tangible or intangible;

b. In a format readable by humans or by a computer;

c. In transit between computers or within a computer network or between any devices which comprise a computer; or

d. Located on any paper or in any device on which it is stored by a computer or by a human; and

4. Computer services.

A person *"uses"* a computer or computer network when he:

1. Attempts to cause or causes a computer or computer network to perform or to stop performing computer operations;

2. Attempts to cause or causes the withholding or denial of the use of a computer, computer network, computer program, computer data or computer software to another user; or

3. Attempts to cause or causes another person to put false information into a computer.

A person is *"without authority"* when he has no right or permission of the owner to use a computer, or, he uses a computer in a manner exceeding such right or permission.

§ 18.2-152.3. Computer fraud.

Any person who uses a computer or computer network without authority and with the intent to:

1. Obtain property or services by false pretenses;

2. Embezzle or commit larceny; or

3. Convert the property of another shall be guilty of the crime of computer fraud. If the value of the property or services obtained is $200 or more, the crime of computer fraud shall be punishable as a Class 5 felony. Where the value of the property or services obtained is less than $200, the crime of computer fraud shall be punishable as a Class 1 misdemeanor.

§ 18.2-152.4. Computer trespass; penalty.

Any person who uses a computer or computer network without authority and with the intent to:

1. Temporarily or permanently remove computer data, computer programs, or computer software from a computer or computer network;

2. Cause a computer to malfunction regardless of how long the malfunction persists;

3. Alter or erase any computer data, computer programs, or computer software;

4. Effect the creation or alteration of a financial instrument or of an electronic transfer of funds;

5. Cause physical injury to the property of another: or

6. Make or cause to be made an unauthorized copy, in any form, including, but not limited to, any printed or electronic form of computer data, computer programs, or computer software residing in, communicated by, or produced by a computer or computer network shall be guilty of the crime of computer trespass, which shall be punishable as a Class 1 misdemeanor. If such act is done maliciously and the value of the property damaged is $2,500 or more, the offense shall be punishable as a Class 6 felony.

§ 18.2-152.5. Computer invasion of privacy.

A. A person is guilty of the crime of computer invasion of privacy when he uses a computer or computer network and intentionally examines without authority any employment, salary, credit or any other financial or personal information relating to any other person. "Examination" under this section requires the offender to review the information relating to any other person after the time at which the offender knows or should know that he is without authority to view the information displayed.

B. The crime of computer invasion of privacy shall be punishable as a Class 3 misdemeanor.

§ 18.2-152.6. Theft of computer services.

Any person who willfully uses a computer or computer network, with intent to obtain computer services without authority, shall be guilty of the crime of theft of computer services, which shall be punishable as a Class 1 misdemeanor.

§ 18.2-152.7. Personal trespass by computer.

A. A person is guilty of the crime of personal trespass by computer when he uses a computer or computer network without authority and with the intent to cause physical injury to an individual.

B. If committed maliciously, the crime of personal trespass by computer shall be punishable as a Class 3 felony. If such act be done unlawfully but not maliciously, the crime of personal trespass by computer shall be punishable as a Class 1 misdemeanor.

§ 18.2-152.8. Property capable of embezzlement.

For purposes of § 18.2-111, personal property subject to embezzlement shall include:

1. Computers and computer networks;

2. Financial instruments, computer data, computer programs, computer software and all other personal property regardless of whether they are:

a. Tangible or intangible;

b. In a format readable by humans or by a computer;

c. In transit between computers or within a computer network or between any devices which comprise a computer; or

d. Located on any paper or in any device on which it is stored by a computer or by a human; and

3. Computer services.

§ 18.2-152.9. Limitation of prosecution.

Notwithstanding the provisions of § 19.2-8, prosecution of a crime which is punishable as a misdemeanor pursuant to this article must be commenced before the earlier of (i) five years after the commission of the last act in the course of conduct constituting a violation of this article or (ii) one year after the existence of the illegal act and the identity of the offender are discovered by the Commonwealth, by the owner, or by anyone else who is damaged by such violation.

§ 18.2-152.10. Venue for prosecution.

For the purpose of venue under this article, any violation of this article shall be considered to have been committed in any county or city:

1. In which any act was performed in furtherance of any course of conduct which violated this article;

2. In which the owner has his principal place of business in the Commonwealth;

3. In which any offender had control or possession of any proceeds of the violation or of any books, records, documents, property, financial instrument, computer software, computer program, computer data, or other material or objects which were used in furtherance of the violation;

4. From which, to which, or through which any access to a computer or computer network was made whether by wires, electromagnetic waves, microwaves, or any other means of communication;

5. In which the offender resides; or

6. In which any computer which is an object or an instrument of the violation is located at the time of the alleged offense.

§ 18.2-152.11. Article not exclusive.

The provisions of this article shall not be construed to preclude the applicability of any other provision of the criminal law of this Commonwealth which presently applies or may in the future apply to any transaction or course of conduct which violates this article, unless such provision is clearly inconsistent with the terms of this article.

§ 18.2-152.12. Civil relief; damages.

A. Any person whose property or person is injured by reason of a violation of any provision of this article may sue therefor and recover for any damages sustained, and the costs of suit. Without limiting the generality of the term, "damages" shall include loss of profits.

B. At the request of any party to an action brought pursuant to this section, the court may, in its discretion, conduct all legal proceedings in such a way as to protect the secrecy and security of the computer, computer network, computer data, computer program and computer software involved in order to prevent possible recurrence of the same or a similar act by another person and to protect any trade secrets of any party.

C. The provisions of this article shall not be construed to limit any person's right to pursue any additional civil remedy otherwise allowed by law.

D. A civil action under this section must be commenced before expiration of the time period prescribed in § 8.01-40.1.

§ 18.2-152.13. Severability.

If any provision or clause of this article or application thereof to any person or circumstances is held to be invalid, such invalidity shall not affect other provisions or applications of this article which can be given effect without the invalid provision or application, and to this end the provisions of this article are declared to be severable.

§ 18.2-152.14. Computer as instrument of forgery.

The creation, alteration, or deletion of any computer data contained in any computer or computer network, which if done on a tangible document or instrument would constitute forgery under Article 1 (§ 18.2-168 et seq.) of Chapter 6 of this Title, will also be deemed to be forgery. The absence of a tangible writing directly created or altered by the offender shall not be a defense to any crime set forth in Article 1 (§ 18.2-168 et seq.) of Chapter 6 of this Title if a creation, alteration, or deletion of computer data was involved in lieu of a tangible document or instrument.

Electronic Data: Privacy, Security and Confidentiality

Ronald J. Palenski
Gordon & Glickson P.C.
Washington, D.C. 20006
McLean, Virginia 22182

Steve A. Mandell
The Mandell Law Firm
A Professional Corporation
Vienna, Virginia 22182

I. INTRODUCTION AND PROBLEM STATEMENT

Computer and communications technologies can provide organizations effective means of establishing instant communication among staff, suppliers, and customers around the world. Internet and Intranet use is growing at a staggering pace, and organizations that fail to take advantage of the medium will be at a disadvantage to competitors who do. As organizations rush to become part of the "Information Age," however, the risks must be considered, and policies must be implemented to avoid them. After all, an organization's computerized data -- from proprietary research and marketing plans, to employee files, and customer lists -- may be little more than a hacked or stolen password away from prying eyes or one download away from destruction.

A 1995 survey, sponsored by the American Society for Industrial Security, found a 323% increase in the reported incidents of information theft per month as compared with the 1992 survey. Of the 700 incidents reported for 1993, 1994, and the first seven months of 1995 by the 325 responding companies, 28% involved computer hackers. In most cases, the incidents involving computer hackers were reported by high tech companies, i.e., computer, software, pharmaceutical, telecommunications, electronics and aerospace companies. Richard J. Heffernan and Dan T. Swartwood, Trends In Intellectual Property Loss Survey, American Society for Industrial Security (March 1996).

An InformationWeek/Ernst & Young survey of 1,290 information systems executives in October 1995 found that one in five companies suffered break-ins and attempted break-ins by way of the Internet. Moreover, the actual number of break-ins is probably much higher, since only one-half of the surveyed managers felt confident that they would be able to detect a system break-in via the Internet. Bob Violano, Internet Security -- Your Worst Nightmare, InformationWeek, Feb. 19, 1996 at 34.

Otherwise, criminals -- including corporate insiders and organized gangs -- are stealing personal computers, computer chips, and corporate data. According to the Chubb Group of insurance companies, losses from chip-theft-related-claims cost it some $20 million in 1995, up from $15 million the year before. Bob Violano, High-Tech Crime: Stop, Thief!, InformationWeek, Mar. 18, 1996 at 36. Who is doing the stealing? It depends. According to 200 U.S. corporate security chiefs surveyed by Michigan State University in East Lansing, chips and components are typically taken by organized gangs. Corporate data, on the other hand, by insiders. Nearly 60% of the survey respondents indicated that their organizations' employees have stolen or tried to steal product information while more than 55% believe their employees have stolen marketing information. Id.

II. PROTECTED ASSETS

A. Corporate Information Systems.

1. Computer hardware including mainframes, peripherals, desktop and laptop computers.

2. Computer hardware components including processing and memory chips.

3. Computer software.

B. Strategic Corporate Information.

1. Trade secrets including technical "know-how".

2. Customer lists, business plans, and financial records.

3. Medical and personnel records.

C. Corporate Reputation.

III. THREATS TO CORPORATE COMPUTER SYSTEMS SECURITY

A. External Threats.

1. Hackers.

2. Physical theft.

3. Viruses, worms, and other programmed pests.

a. Viruses.

A "virus" is a sequence of computer instructions inserted into a program such that when the program is run, the viral instructions are also executed.

b. Worms.

A "worm" is not dependent on a host program for its existence. "Worms are programs that can run independently and travel from machine to machine across network connections; worms may have portions of themselves running on different machines." Then-Cornell graduate student Robert T. Morris shut down computers lined by the Internet in November 1988 by disseminating a worm, not a virus as widely reported.

c. Trojan horses.

A "Trojan Horse" is malevolent program code disguised as a legitimate application, such as a word processor, spreadsheet, or database management program.

d. Logic bombs.

"Logic bombs" are embedded programming which may destroy or alter data or cause machine halts when a certain set of conditions are met. The triggering condition is often a date, such as April 1 or Friday the 13th. For an extensive technical discussion of computer pests, see Eugene Spafford, Katherine Heaphy, and D. Ferbrache, Computer Viruses: Dealing with Electronic Vandalism and Programmed Threats, ADAPSO (1989).

B. Internal Threats.

1. Dishonest or disgruntled employees or former employees.

2. Unauthorized systems users.

3. Careless or ignorant system users.

4. Viruses, worms, and other programmed pests.

IV. THREATS TO PERSONAL PRIVACY AND SECURITY

A. Sexual Harassment and Stalking in an On-Line Era.

B. E-mail Monitoring.

V. PREVENTION AND REMEDIES

A. Technical.

1. Firewalls.

"Firewall" refers to software and hardware that form a protective gateway or wall between an individual PC or an organization's internal network and the Internet. There are two basic types of firewalls: screening routers and bastion hosts.

a. Screening routers.

Data is transmitted on the Internet in packets, and screening routers act as an advance switch which can allow the packet into a network or reject it. "Screening routers" can be configured to filter certain source addresses, such as those that denote educational institutions, which are notorious sources of hackers. Screening routers are low-cost and a good starting point for companies without high security requirements.

b. Bastion hosts.

A "bastion host" firewall forces all traffic between the Internet and the protected network to pass through the bastion host for detailed analysis. Users are required to "unlock" access to the network by using a smart card or a challenge-response calculator that identifies them. Most bastion hosts include burglar alarms to bring unusual events to the immediate attention of the system administrators. They also provide excellent audit capabilities since all traffic must first past through them. Whilc bastion hosts provide a higher level of security than router based firewalls, they are much more costly and provide less flexibility. Sec generally Marcus J. Ranum, Great Walls of Fire, Security Managemcnt, July 1995, at 131.

2. Encryption.

Digital encryption makes electronic data unreadable except to those authorized to decrypt the information. Thcre are two basic types of encryption in use today: symmetrical-key encryption (also called single-key or private-key encryption) and public-key encryption.

a. Symmetrical-key encryption.

Symmetrical-key encryption uses the same key to encrypt and decrypt data. This method creates the problem of communicating the encryption key without compromising it.

b. Public-key encryption.

Public-key encryption involves mathematical algorithms that factor large numbers and create two numbers or "keys". One of the keys creates an encrypted message from plain text, and the other recovers the plain text from the encrypted version. The key held by the person using the technique is not disclosed to anyone and is thus referred to as the "private key". The other key is disclosed publicly. When a sender wishes to send a private message to an addressee, the sender uses the public key belonging to the addressee to encrypt the message. Only the addressee can discern the contents by unlocking the message with his or her private key. The level of security provided by encryption increases exponentially as the key length is increased. See Jon Kaplan, Unscrambling the Secret of Encryption, Security Management, February 1995, at 67.

3. Digital signatures.

Variation of public-key encryption which involves the creation of "digital signatures" for each individual that cannot be duplicated. Digital signatures would prevent people from impersonating others as they exchange information and buy goods over computer networks. For an extensive discussion of digital signatures, see generally Digital Signature Guidelines, Information Security Committee, A.B.A. Sec. Sci. and Tech. (1996).

4. Passwords.

5. Virus-checking software.

6. System backups.

B. Legal.

1. Federal and state computer crime laws.

a. Federal Computer Fraud and Abuse Act (18 U.S.C. § 1030).

The Computer Fraud and Abuse Act makes it a crime to access certain computers without authorization or in excess of authorization and thereby to cause certain results. The Secret Service has the authority to investigate offenses and violations are punishable by substantial fines and imprisonment up to 10 years imprisonment (up to 20 years for a repeat offense). The Act covers computers used by financial institutions or the United States government and any computer which is "one of two or more computers used in committing the offense, not all of which are located in the same state."

The following is a list of the offenses criminalized by the statute:

(1) Obtaining classified information pertaining to national defense or foreign relations or restricted atomic energy data with intent or reason to believe

that the information so obtained is to be used to the injury of the United States or to the advantage of any foreign nation. 18 U.S.C. § 1030(a)(1).

(2) Obtaining information contained in a financial record of a financial institution, of a credit card issuer, or of a consumer reporting agency relating to a consumer. 18 U.S.C. § 1030(a)(2).

(3) Accessing a computer of a department or agency of the United States and adversely affecting the government's use of such computer. 18 U.S.C. § 1030(a)(3).

(4) Furthering an intended fraud by accessing a covered computer unless the fraud consists only of the use of the computer. 18 U.S.C. § 1030(a)(4).

(5) Transmitting program, information, code, or command intending that it will damage a computer system, data, or program or delay or deny the use of such program and such conduct causes loss or damage aggregating more than $1,000 during any one-year period. 18 U.S.C. § 1030(a)(5)(A).

(6) Transmitting data, information, code, or program that actually or potentially modifies or impairs medical information. 18 U.S.C. § 1030(a)(5)(A).

(7) Transmitting program, information, code, or command with reckless disregard of a substantial and unjustifiable risk that the transmission will damage the operation of computer system, information, data, or program or deny or delay the use of a computer system and thereby cause more than $1,000 aggregate damage during a one-year period, or modify or impair medical information. 18 U.S.C. § 1030(a)(5)(B).

(8) Knowingly and with intent to defraud, trafficking in any computer access passwords. 18 U.S.C. § 1030(a)(6).

See United States v. Morris, 928 F.2d 504 (2d Cir. 1991). The Second Circuit affirmed the conviction under the Computer Fraud and Abuse Act of Robert T. Morris, a Cornell University graduate student, who released a computer worm onto the Internet. Morris had placed the rogue program onto the Internet from a computer at the Massachusetts Institute of Technology. The Court made two important points in the interpretation of the Act. First, intentional access is sufficient under the Act; the government need not prove intent to cause damage or injury. Second, the unauthorized access element is satisfied when a computer to which one has authorized access is used for an unauthorized purpose.

b. Virginia Computer Crimes Act (Va. Code Ann. §§ 18.2-152.1 to -152.14).

The Virginia Computer Crimes Act makes it a crime to commit:

(1) Computer fraud.

To use a computer or computer network without authority and with the intent to obtain property or services by false pretenses; to embezzle or commit larceny, or to convert the property of another. Va. Code Ann. § 18.2-152.3.

(2) Computer trespass.

To use a computer or network without authority and with the intent to remove, temporarily or permanently, computer data, programs or software; cause computer malfunction; alter or erase any computer data or computer programs; effect the creation or alteration of financial instrument or an electronic transfer of funds; cause physical injury to property of another, or make unauthorized copy of computer programs or software. Va. Code Ann. § 18.2-152.4.

(3) Computer invasion of privacy.

To use a computer or network to intentionally examine, without authority, any employment, salary, credit or other financial or personal information relating to any other person. Va. Code Ann. § 18.2-152.5.

(4) Theft of computer services.

To willfully use a computer or computer network with intent to obtain computer services without authority. Va. Code Ann. § 18.2-152.6.

(5) Personal trespass by computer.

To use a computer or computer network without authority and with intent to cause physical injury to another. Va. Code Ann. § 18.2-152.7.

(6) Embezzlement.

To commit embezzlement with respect to computers and computer networks, financial instruments, computer data, and computer programs, and computer services. Va. Code. Ann. § 18.2-152.8.

Violations are punishable by up to 10 years imprisonment and up to $2,500 fine. Moreover, civil relief, in addition to any other civil remedy allowed, is available to any person whose property or person is injured by reason of a violation of the Virginia Computer Crimes Act. Va. Code Ann. § 18.2-152.12.

See Rosciszewski v. Arete Assocs., 1 F.3d 225 (4th Cir. 1993) (enumerating the elements necessary to show a violation of Section 18.2-152.3 of the Act).

2. Federal and state intellectual property laws.

a. Copyright (17 U.S.C. § 101 et seq.).

Federal copyright law is an important form of intellectual property law for protecting computer programs and other data.

(1) Subject matter.

Copyright protection subsists in original works of authorship fixed in a tangible medium of expression, now known or later developed, from which the works can be perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device. Protected works of authorship include: literary works (including computer programs and digital databases), musical works, motion pictures and other audiovisual works, pictorial works, and sound recordings. 17 U.S.C. § 102. A "computer program" is defined as "a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result." 17 U.S.C. § 101.

(2) Exclusive rights.

Copyright protection is limited to the expression of ideas and not processes, procedures, methods of operation, and the like. 17 U.S.C. § 102(b). With certain exceptions, copyright affords the owner of copyright the exclusive rights to: (i) reproduce the work in copies or phono records; (ii) prepare derivative works based on the copyrighted works; (iii) distribute copies or phono records of the copyrighted work to the public by sale or other transfer of ownership, or by rental, lease, or lending;

(iv) perform the copyrighted work publicly; and (v) display the copyrighted work publicly. 17 U.S.C. § 106.

(3) Infringement.

Anyone who violates any of the exclusive rights of the copyright owner, as set forth in 17 U.S.C. §§ 106 through 118, is an infringer of copyright. "Anyone" includes any state, instrumentality of a state, and any officer or employee of a state or instrumentality thereof, acting in his or her official capacity. 17 U.S.C. § 501.

(4) Remedies.

Civil remedies for infringement include: temporary and permanent injunctions; impounding and disposition of infringing articles; actual or statutory damages (at plaintiff's election); and, depending upon whether the copyright was registered before the infringement, costs and attorneys fees. 17 U.S.C. §§ 502-505.

(5) Criminal offenses.

Any person who infringes a copyright willfully and for purposes of commercial advantage or private financial gain shall be punished as provided in 18 U.S.C. § 2319 (fines of up to $250,000 and imprisonment for up to 10 years). 17 U.S.C. § 506.

(6) Fair use.

Fair use is an affirmative defense, based in principles of equity, to an action for copyright infringement. It is potentially available with respect to all unauthorized uses of works in all media. In determining whether a use is fair, a court must consider at least the following: (i) the purpose and character of the use, including whether it is of a commercial nature or is for nonprofit, educational purposes; (ii) the nature of the copyrighted work (e.g., highly fanciful or factual); (iii) the amount and substantiality of the portion used in relation to the copyrighted work as a whole; and (iv) the effect of the use upon the potential market for or value of the copyrighted work. 17 U.S.C. § 107.

The jurisprudence of fair use is quite extensive. Meanwhile, the scope of fair use is an issue of ongoing debate in the context of the Internet where information in digital format is easily, quickly, and perfectly exchanged.

(7) Copyright in on-line environments.

On-line environments, including the Internet, Intranets, bulletin board systems, and the like, may involve a variety of intermediaries between a content originator and the ultimate information user. This has created a threat of potential liability for various intermediaries for copyright infringement, especially where the intermediaries have deep pockets or are more amenable to personal jurisdiction than the originator of the allegedly infringing work. See generally Intellectual Property and the National Information Infrastructure, Report of the Working Group on Intellectual Property Rights (Sep. 5, 1995). See also H.R. 2441 and S. 1284, now pending before the U.S. Congress.

(a) Direct v. indirect infringement.

Direct infringement. Section 501 of the Copyright Act prohibits a person who is not the copyright owner from engaging in any of the activities enumerated in Section 106 as exclusive rights of the copyright owner, under threat of civil liability as a *direct infringer*.

Indirect and vicarious liability. The Copyright Act contains no specific statutory authority for finding liability against a party for copyright infringement committed by another, but courts have developed the theories under which persons who are not themselves engaging in infringing activities are liable for copyright infringement based on their connection to another person's violation. See, e.g., Sony Corp. v. Universal City Studios, Inc., 464 U.S. 417 (1984). Two distinct theories exist:

i) *Vicarious liability* -- focuses on the relationship between the direct and the third-party infringer. Two elements must be satisfied. First, the third-party infringer must have the right and ability to control and supervise the activities of the infringing party. Second, the third-party must have an obvious and direct financial interest in the activities of the direct infringer. See Shapiro, Bernstein & Co. v. H.L. Green Co., 316 F.2d 304 (2d Cir. 1963).

ii) *Contributory copyright infringement* -- focuses on knowledge of and contribution to the illegal act and stems from the tort doctrine of enterprise liability. See Gershwin Publishing Corp. v. Columbia Artists Management, Inc., 443 F.2d 1159, 1162 (2d Cir. 1971) ("One who, with knowledge of the infringing activity, induces, causes or materially contributes to the infringing conduct of another, may be held liable as a "contributory" infringer.")

(8) Recent decisions.

(a) Playboy Enterprises v. Frena, 839 F. Supp. 1552 (M.D. Fla. 1993). The Court found the defendant bulletin board operator liable as a *direct* copyright infringer. The issue before the Court was whether the bulletin board operator was liable for the acts of users who had uploaded and downloaded approximately 170 of Playboy's copyrighted photographs. Despite the defendant's defense of lack of knowledge of the infringing activity, the court found the operator liable as a direct infringer on the ground that providing access to the computer bulletin board was equivalent to "distributing" and "displaying" the infringing photos. The Court emphasized the fact that the works were clearly identified as Playboy's works and a large number of photographs were available on the bulletin board.

(b) Sega Enterprises v. Maphia, 857 F. Supp. 679 (N.D. Cal. 1994). The Court held defendant bulletin board operator liable for copyright infringement based on the theory of third-party liability. The issue before the Court was whether the defendant bulletin board operator was liable for the acts of its users who had uploaded and downloaded Sega's copyrighted video games. The court noted that, in the instances where the defendants did not themselves upload or download copyrighted games, liability could still be found under the third-party liability theory of contributory copyright infringement, "even if Defendants do not know exactly when games will be uploaded to or downloaded from the bulletin board, their role in the copying, including the provision of facilities, direction, knowledge and encouragement, amounts to contributory copyright infringement." Id. at 684.

(c) Religious Technology Center v. Netcom On-Line Services, Inc., 907 F. Supp. 1361 (N.D. Cal. 1995). Defendant on-line service provider found neither directly, contributorily, nor vicariously liable for copyright infringement of the works of Church of Scientology founder, L. Ron Hubbard. With respect to claims of direct infringement, the Court analogized Netcom's role to that of a photocopy machine owner who permits its use by the public. ("Although copyright is a strict liability statute, there should still be some element of volition or causation which is lacking where a defendant's system is merely used to create a copy by a third party.") The Court similarly found that Netcom did not have the requisite knowledge of the infringing activity nor did it participate in it. As to Netcom's vicarious liability, although the Court did note a genuine issue of fact with regard to Netcom's ability to control the infringement, it did not find the requisite financial benefit sufficient to support a claim of vicarious liability.

(d) Fonovisa v. Cherry Auction, CV 94-15717 (9th Cir. Jan. 25, 1996). Although not an on-line services case, the Ninth Circuit's recent decision has major implications with respect to the vicarious liability of on-line services providers, BBS operators, and others. Here, a music company was allowed to pursue claims of vicarious liability for copyright infringement against the operator of a "swap meet" where third-party vendors regularly sold counterfeit sound recordings, with knowledge and failure to act on the part of the swap meet owner.

b. Trademark.

Trademark law is another important form of intellectual property protection in on-line environments.

(1) Subject matter.

Trademark protection is designed to protect the name, design, or other indicia of origin under which a seller distinguishes his goods and services from those of another. The Lanham Act defines the term "trademark" as including any word, name, symbol, or device, or any combination thereof: (1) used by a person, or (2) which a person has a bona fide intention to use in commerce and applies to register on the principal register established by this Act, to identify and distinguish his or her goods, including a unique product, from those manufactured or sold by others and to indicate the source of the goods, even if that source is unknown. 15 U.S.C. § 1127.

(2) Limited protection.

Trademark protection is limited to those marks which are inherently distinctive or have acquired secondary meaning, i.e., invokes a connection in the consumer's mind between the mark and the provider of the goods or service. Marks are often classified in categories of generally increasing distinctiveness, including (1) generic; (2) descriptive; (3) suggestive; (4) arbitrary; or (5) fanciful. See Two Pesos, Inc. v. Taco Cabana, Inc., 112 S. Ct. 2753, 2757 (1992). Marks that are merely descriptive of a product do not inherently qualify, but they may acquire the distinctiveness that causes them to represent information about a particular source of a product by acquiring a secondary meaning.

(3) Infringement.

Trademark infringement occurs when someone other than the trademark owner uses the same or a confusingly similar term on the same or closely related goods or services in the same geographical area, or in some circumstances, within a natural area of expansion.

(4) Remedies.

Civil remedies are available under the federal law and most state laws, including: (a) an injunction against future infringement; (b) the infringer's profits; (c) damages for past infringement suffered by the owner of the mark; (d) destruction of all materials bearing the infringing mark; and (e) the costs of the action and, in exceptional cases, reasonable attorney's fees. See Va. Code Ann. § 59.1-88 (1992). In addition, some states have criminal penalties for certain forms of trademark infringement. See Va. Code Ann. § 59.1-89 (Supp. 1992) (provides that any person who knowingly and intentionally infringes a trademark registered under the Virginia Trademark and Service Mark Act, shall be guilty of a Class 2 misdemeanor).

(5) Trademark law applied in on-line environments.

(a) Domain names.

Trademark protection has been applied in the NII context to domain names. Domain names are the alphanumeric address of an Internet user usually consisting of a word or words such as an individual's, organization's, or company's name, a brand name or trademark, or any other word commonly associated with a particular user. The domain name includes a three-letter abbreviation indicating the user's type of organization. (*.com* - commercial; *.edu* - educational; *.gov* - government; *.org* - nonprofit organizational; *.net* - network; *.mil* - military).

Approval and registration of Internet domain names is administered by Internet National Information Center (InterNIC), under a cooperative agreement with the National Science Foundation. Domain names are granted and registered free of charge by InterNIC on a first come first served basis. As of July 1995, 70,000 commercial names were registered. InterNIC does not run trademark searches on domain names submitted for approval and registration, it merely checks its records to ensure that an identical domain name has not already been issued.

New policy. An applicant for a domain name registration must: (a) declare it has the right to use the name; (b) declare a bona fide intention to use the name regularly on the Net; and (c) declare that registration is not sought for any unlawful purpose, including trademark infringement. See NSI Policy Statement, published on July 28, 1995 (ftp://rs. Internic, net/policy/internic/ internic-domain-1.txt.) .

(b) Trademarks.

The on-line environment is fertile ground for trademark and service mark infringement. While there is scant decisional law specifically involving on-line infringement, this is one area where the traditional legal standards fit nicely in the on-line environment. See Sega Enterprises, Ltd. v. Maphia, 857 F.Supp. 679 (N.D. Cal. 1994); see also discussion supra of Playboy Enterprises Inc. v. Frena, 839 F.Supp. 1552 (M.D. Fla. 1993).

(6) Recent decisions.

(a) MTV Networks v. Curry, 867 F. Supp. 202 (S.D.N.Y. 1994). MTV, owned by Viacom, sued Adam Curry, a former MTV video-jockey for trademark infringement and breach of contract, for use of the MTV trademark in Curry's Internet domain name. Curry, while employed by MTV, had registered the domain name "mtv.com" in his own name. With MTV's knowledge and, apparently, blessing, Curry conducted an on-line "talk show" via the Internet using the mtv.com domain name. When Curry left MTV he continued to use the mtv.com address. The case was decided on breach of contract grounds, with Curry agreeing to no longer use the domain name.

(b) In re Arbitration Between Stanley H. Kaplan Educational Center, Ltd. v. The Princeton Review Management Corp., No. 13-199-00145 94. Princeton Review registered the domain name "kaplan.com." Princeton's competitor in the standardized test preparation business, Stanley Kaplan Company, brought suit alleging trademark infringement, false and misleading advertising and unfair competition. The parties submitted to arbitration, and the arbitrator ordered Princeton to: (i) notify the InterNIC that it was relinquishing all rights to the "kaplan.com" domain name; (ii) cause the cancellation or revocation of its prior registration of the name; and (iii) request InterNIC to transfer the name to Kaplan. The arbitrator declined to award damages or attorney's fees on the ground that there was an inadequate showing of actual damage or intentional deception or bad faith by Princeton.

(c) Hasbro, Inc. v. Internet Entertainment Group, Ltd., 1996 WL 84853 (W.D. Wash. Feb. 9, 1996). Hasbro, Inc. is the owner of the "CANDYLAND" trademark which has been registered on the Principal Register of the U.S. Patent and Trademark Office since 1951. Defendants used the name CANDYLAND to identify a sexually explicit Internet site by using the domain name "candyland.com". Hasbro sued the defendants for trademark infringement and sought a preliminary injunction. The Court found that Hasbro demonstrated: (i) a likelihood of prevailing on its claims; (ii) that the defendants' use of the CANDYLAND name in connection with their Internet site was causing irreparable injury; and, (iii) that the harm to Hasbro outweighed any inconvenience that the defendants would experience if they are required to stop using the CANDYLAND name. Consequently, the Court preliminarily enjoined the defendants against further use of the name.

(d) Playboy Enterprises Inc. v. Frena, 839 F.Supp. 1552 (M.D. Fla. 1993). The Court found trademark infringement in the unauthorized uploading to a computer bulletin board and subsequent distribution of plaintiff's copyrighted photographs which were identified by the trademarks "PLAYBOY" or "PLAYMATE."

c. Trade secrets.

(1) Trade secret definitions.

Generally speaking, a valid trade secret exists only if it is substantially secret within the trade secret owner's industry. Absolute secrecy is not required but if the trade secret is widely used within the industry, it is less likely that it can be protected as a property right. Whether secrecy exists is a factual question. States have adopted either of two "trade secret" definitions below or some variation thereof.

(a) First Restatement of Torts § 757.

A trade secret is ". . . any formula, pattern, device or compilation of information which is used in one's business, which gives him an opportunity to obtain an advantage over competitors who do not know or use it. A trade secret may be a formula for a chemical compound; a process or manufacturing, treating or preserving materials; a pattern for a machine or other device; or a list of customers."

(b) The Uniform Trade Secrets Act defines a "trade secret" as the whole or any portion or phase of any scientific or technical information, design, process, procedure, formula, improvement, confidential business or financial information, listing of names, addresses, or telephone numbers, or other information relating to any business or profession which is secret and of value.

(2) Virginia Trade Secret Statute (Va. Code Ann. §§ 59.1-336 to -343).

"Trade secret" means information, including but not limited to, a formula, pattern, compilation, program, device, method, technique, or process, that: (1) derives independent economic value, actual or potential, from not being generally known or ascertainable by others who can obtain economic value from its disclosure or use, and (2) is the subject of efforts that are reasonable under the circumstances to maintain its secrecy. The Act provides for injunctive relief where trade secret misappropriation is threatened or actual.

(3) Recent decisions.

(a) Dionne v. Southeast Foam Converting & Packaging, Inc., 240 Va. 297 (1990). Defendant's father incorporated plaintiff corporation as a family-owned company producing foam packing products. The company began developing new manufacturing processes for compressed foam and employed defendant son and his brother. In the aftermath of a bitter family quarrel, all family members received stock in the corporation and signed a confidentiality agreement. However, harmony was not restored and the defendant's employment was terminated. Defendant planned to start a new business manufacturing a foam material for use in the inner packaging industry. When family learned of these plans, plaintiff filed its bill of complaint. The Chancellor held that the manufacture of the particular product by the plaintiff corporation was a trade secret and that defendant's conduct constituted misappropriation. The Chancellor entered an injunction. The Circuit Court affirmed but noted that when the Durafoam process ceases to be a trade secret, defendant may petition the Court to modify and limit the injunction.

(b) American Sales Corporation v. Adventure Travel, Inc., 862 F.Supp. 1476 (E.D. Va. 1994). The Court held that American Sales Corporation ("ASC") was entitled to reasonable royalty damages of $22,500 from Adventure Travel, Inc. ("ATI") for the misappropriation of a trade secret under the Virginia Trade Secrets Act ("VTSA"). ASC sells a collection of discount services for a membership fee. Pursuant to a contract, ATI provided discount travel services to ASC members. In order for ATI to perform its obligations under the contract, ASC provided it with a member list, updated daily, but stressed in writing the importance of keeping the list confidential and prohibited ATI from using the list for its own gain. After the contract was terminated, ATI created its own marketing company offering discount services very similar to those offered by ASC and began soliciting ASC members. ATI was found liable for the misappropriation of trade secrets through summary judgment and the parties presented the issue of damages at trial. The Court ordered damages in an amount of $22,500 to be paid to ASC based upon what ATI would have reasonably paid for a license to use the list.

3. Federal and state privacy laws.

a. Electronic Communications Privacy Act (18 U.S.C. §§ 2510-2521, 2701-2710).

The Electronic Communications Privacy Act ("ECPA"), enacted in 1986, codifies the warrant requirements for the interception of electronic communications by government officials and creates privacy protections for stored electronic messages. Title I of the ECPA covers acquisition and disclosure of communications streams. Title II covers acquisition and disclosure of stored information. Title III covers the acquisition and disclosure of transactional information. Subsequent amendments have added protection in the area of videotape rental records and regulation addressing transponders (mobile tracking devices).

(1) Criminal prosecution.

The ECPA is used to prosecuting unauthorized access and certain disclosures of electronic communications. Anyone who intentionally accesses without authorization a facility through which an electronic communication service is provided or intentionally exceeds an authorization to access that facility and "thereby obtains, alters, or prevents authorized access to a wire or electronic communication while it is in electronic storage of such system" is subject to a fine of up to $250,000 and imprisonment for up to one year for a first offense or two years if for a repeated offense. 18 U.S.C. § 2701. Where criminal intent is not proven, the fine drops to $5,000 and the imprisonment term drops to six months. 18 U.S.C. §2701(b)(1)(b).

(2) Civil remedies.

Civil remedies are authorized against anyone who violates Title I of the ECPA. 18 U.S.C. § 2520.

(3) Prosecutorial use.

The ECPA immunizes providers of wire and electronic communication services, their officers, employees, agents, landlords, custodians, and other persons who are providers either with a court order directing assistance to law enforcement authorities by disclosing covered wire or electronic communications or by attorney general certifications. 18 U.S.C. § 2518(7).

(4) E-mail monitoring by employers.

The scope of ECPA with respect to employer monitoring of employee E-mail is unclear. The ECPA does permit the provider of an electronic communication service to intercept messages for the "protection of the service's property or rights." The Act goes on to state, however, that the provider shall not use service observing or random monitoring except for mechanical and service quality control checks. 18 U.S.C. § 2511. At a minimum and to negate an expectation of privacy claims, employers should advice employees, in employee manuals and on-screen, that corporate computer systems are business instrumentalities to be used for business purposes and that the information stored and transmitted on them is accessible by the organization's system administrator.

(a) Strauss v. Microsoft Corp., No. 91 Civ. 5928 (S.D.N.Y. June 1, 1995). Judge held that E-mail messages containing derogatory references to certain employees were admissible in sex discrimination cases under Title VII of the 1964 Civil Rights Act.

b. Fair Credit Reporting Act ("FCRA") (15 U.S.C. §§ 1681 - 1681t).

The FCRA regulates the dissemination of consumer credit reports by consumer reporting agencies and is the most far-reaching of the federal privacy laws. Both consumer-reporting agencies and users of consumer reports are subject to civil liability for willful noncompliance with the

FCRA. This includes liability for actual damages sustained by the consumer, punitive damages, and legal and attorney's fees. In the event of negligent noncompliance, the consumer may recover actual damages plus legal and attorneys' fees. The statute of limitations for bringing an action is two years from the date liability arises. Unauthorized disclosures of consumer reports by consumer reporting agencies are subject to criminal penalties, including a fine of up to $5,000, imprisonment of up to one year, or both.

c. Right to Financial Privacy Act (12 U.S.C. §§ 3401 - 4322).

The Right to Financial Privacy Act limits the right of the federal government to obtain financial records from financial institutions. The government must provide a formal written statement that includes the nature of the records sought and the purposes of the disclosure. A copy of the request must be sent to the financial institution's customer, who has the right to challenge access by the government. There are exceptions which permit a financial institution to provide specific information when it suspects that a law or regulation has been violated. The information which can be provided, though, is limited to the name of the person involved (or the identifying account information) and the nature of the suspected illegal activity.

Federal agencies and financial institutions are civilly liable to customers for the wrongful disclosure of financial information. An aggrieved customer may recover $100 without regard to the volume of records involved, actual and punitive damages, court costs and attorneys' fees.

d. Virginia Privacy Protection Act (Va. Code Ann. §§ 2.1-377-386).

The Virginia Privacy Protection Act requires government agencies to take certain procedural steps in connection with the collection, maintenance, use and dissemination of personal information. The Act provides for injunctive relief when an agency has violated the required procedures.

In Virginia it is a crime to intentionally or without authorization or in excess of authorization to examine any employment, salary, credit or other personal information relating to another. Va Code Ann. § 18.2-152.5 (1988).

e. Medical information.

Many states prohibit disclosure of an individual's medical information to third parties without the consent of the individual. Colorado, for example, has criminalized the knowing obtaining of medical information without authorization and with intent to appropriate it for one's own use or for the use of another. Many states also have a Mental Health Act which guarantees the confidentiality of mental health records. See, e.g., Illinois Mental Health and Developmental Disabilities Confidentiality Act, 740 ILCS 110/1 - 110/17 (West 1993). In Virginia, physicians are required to follow the American Medical Association's guidelines regarding the disclosure of an individual's medical information.

f. Common law invasion of privacy.

The common law recognizes four invasion of privacy torts: (i) intrusion upon seclusion; (ii) disclosure of private facts; (iii) portrayal in a false light; and (iv) appropriation of name or likeness.

Virginia law, however, does not recognize the common law causes of action for the invasion of privacy. Instead, the Virginia General Assembly has created a statutory cause of action for the appropriation of another's name or likeness. Va. Code Ann. § 8.01-40 (1987). See Falwell v. Penthouse Int'l. Ltd., 521 F.Supp. 1204 (W.D.Va. 1981) ("The only remedy available for an invasion of privacy in Virginia is statutory. Virginia has never recognized a common law cause of action for invasion of privacy.").

g. Recent decision.

Stern v. Delphi Internet Services Corp., 626 N.Y.S.2d 694 (1995). Radio disk jockey Howard Stern sued Delphi, an Internet access provider, over its use of his picture. Delphi had created an electronic bulletin board on the Internet to debate the merits of Mr. Stern's announced candidacy for governor of New York. To advertise the bulletin board, Delphi ran advertisements in New York Magazine and the New York Post containing a picture of Mr. Stern wearing leather pants that largely exposed his buttocks. Stern brought action under a New York statute providing relief for invasion of privacy. Delphi argued that the use of Mr. Stern's photograph was within the scope of the "incidental use" and "newsworthiness" exceptions to the statute. The court held that particularly in light of Mr. Stern's bid for the governorship the published material fell within the "newsworthiness" exception afforded news agencies under the First Amendment.

5. Common law defamation.

The four key elements of a defamation claim are: (i) a false and defamatory statement about another party; (ii) published to one or more third parties without privilege; (iii) by a publisher who is at least negligent in communicating the information; and (iv) which results in presumed or actual damage. See Chapin v. Greve, 787 F.Supp. 557 (E.D.Va. 1992).

a. Defamation law applied in on-line environments.

In addition to copyright liability, discussed in Section 2a above, intermediaries also face the threat of potential liability for defamation. Accordingly, another major issue in on-line environments involves questions of who is liable as a "publisher" of defamatory material.

b. Recent decisions.

(1) Cubby, Inc. v. CompuServe, 776 F.Supp. 135 (S.D.N.Y. 1991). Court held that CompuServe was not liable for the defamatory statements made on one of the "on-line electronic fora" provided by CompuServe. The Court found that CompuServe was a "distributor" of the materials and thus held to a standard of negligence. Unlike traditional publishers, CompuServe relinquished its editing control to independent organizations. In 1991, CCI, a third-party, agreed to review and edit the contents of the Journalism Forum, including the "Rumorville" bulletin board published by yet another entity--Don Fitzpatrick Assoc. Accordingly, CompuServe did not actually review the contents of Rumorville before placing the newsletter on-line. The Plaintiff, Cubby, Inc., charged CompuServe with making defamatory remarks via Rumorville. CompuServe argued that it was a "distributor" and not a "publisher" and therefore was not liable unless it had reason to know of the bulletin board's content. The Court agreed and compared CompuServe to a traditional news vendor, a bookstore (or distributor) and then applied a negligence standard.

(2) Stratton Oakmont v. Prodigy, 1995 N.Y. Misc. LEXIS 229 (N.Y. Sup. Ct., May 24, 1995). Court held on summary judgment that Prodigy had exercised sufficient editorial control over the content of its system to constitute itself a "publisher" for purposes of a defamation claim. An unidentified user had posted statements on Prodigy's "Money Talk" bulletin board accusing Stratton, an investment banking firm, and its president of criminal and fraudulent acts in connection with an initial private offering ; specifically -- "cult of brokers who either lie for a living or get fired." The Court considered the following facts: (1) marketing information

disseminated by Prodigy stating that Prodigy exercised editorial control over the content of messages posted on its bulletin boards differentiating it from the competition; (2) Prodigy had issued content guidelines requesting users to refrain from posting insulting notes and warning users that Prodigy would remove such notes; (3) Prodigy used a software screening program to filter sexually explicit material.

The Stratton Oakmont decision was expressly overruled by Title V of the Telecommunications Act of 1996 ("No provider or user of an interactive computer service shall be treated as the publisher or speaker of any information provided by another information content provider."). 47 U.S.C. § 230(c).

6. Communications Decency Act of 1995 (47 U.S.C. § 223(a)-(h)).

The Communications Decency Act ("CDA") bans the making of "indecent" or "patently offensive" material available to minors via computer networks. The Act imposes a fine of up to $250,000 and imprisonment for up to two years. The CDA does, however, specifically exempt from liability any person who provides access or connection to or from a facility, system or network that is not under the control of the person violating the Act. Similarly, the CDA states that an employer shall not be held liable for the actions of an employee unless the employee's conduct is within the scope of his or her employment. See 47 U.S.C. § 223(e).

a. American Civil Liberties Union v. Reno, 1996 WL 65464 (E.D. Pa. Feb. 15, 1996). Court granted motion for a temporary restraining order against the defendant, her agents, and her servants enjoining them from enforcing against plaintiffs the provisions of 47 U.S.C. § 223(a)(1)(B)(ii) insofar as they extend to "indecent" materials.

7. State obscenity/indecency/harassment/stalking laws.

On-line environments are not limited by geography; it is, therefore, important for organizations to recognize their potential liability in other jurisdictions. See United States v. Thomas, 74 F.3d 701 (6th Cir. 1996), in which a California couple was convicted in Tennessee of trafficking in obscene materials stored on their computer bulletin board but accessible nationwide. All states have laws governing the distribution of obscene materials. Generally, these prohibit the sale, lending, renting, publication, exhibition or other distribution of materials, with general knowledge of their obscene character and content. In a very few states, obscenity laws cover the dissemination of *tangible* material only, a matter of some importance given the debate in other areas of the law whether computer software or other information in digital format is tangible or intangible. Similarly, many states prohibit the distribution of *electronic or electrical reproductions* of obscene material. It is unclear whether these statutes cover obscene material in digital format.

All states have statutes prohibiting harassment and stalking. "Harassment" statutes typically prohibit the intentional or knowing engaging in a regular course of conduct (which may include sending mail -- including *electronic mail* -- or other written communications) designed to alarm or seriously annoy another. "Harassment" is sometimes included as a subset of "stalking," which is typically defined as the willful, malicious, and repeated harassing of another, or the making of a credible threat, with intent to place another in reasonable fear of death or great bodily harm. . .

All states have statutes prohibiting harassment by telephone, although it appears that most of these statutes contemplate (currently, at least) voice rather than digital communications. In 1995, however, Connecticut amended its law specifically to address harassment by computer network. Conn. Gen. Stat. § 53a-182. For a comprehensive overview of state obscenity, indecency, harassment, and stalking statutes, see generally Ronald J. Palenski, State Laws on Obscenity, Child Pornography and Harassment, Internet, Free Speech and Industry Self-Regulation, Appendix D, Information Technology Association of America (1995).

a. Virginia computer obscenity law amendments.

In 1995, Virginia amended its criminal code with regard to the use of computers and computer networks to distribute obscene material. The new Virginia law, codified at Va. Code Ann. § 18-2.374, prohibits the use or solicitation of a child to make or produce sexually explicit material, which includes *digital* images. The statute further prohibits knowingly participating in the reproduction of such material, including *computer-generated reproduction and electronic transmission*. Finally, the amended statute prohibits the use of *computers, computer networks or bulletin boards, or any other electronic means* to promote sexually explicit material involving a minor.

8. Encryption policy issues.

Under the Arms Export Control Act, the use, sale and export of computer hardware and software embodying "strong" encryption technology is subject to stringent and pervasive regulation by the U.S. Government. Proponents argue that continued regulation is necessary to avoid negative effects on the law enforcement and intelligence communities. Opponents to the regulation argue that American developers of hardware and software are placed at a major disadvantage due to the readily available strong encryption technology from non-U.S. sources. The debate is presently ongoing in the U.S. Congress. Bills were introduced in both the Senate and House in March 1996.

a. Arms Export Control Act (22 U.S.C. §§ 2751-2796d).

The Arms Export Control Act ("AECA") authorized the President to control the import and the export of defense articles and defense services. Pursuant to the AECA, the Secretary of State issued the International Traffic in Arms Regulations, 22 C.F.R. § 120-130 (1994) ("ITAR"). The ITAR lists controlled "defense articles" to include, "the furnishing to foreign persons of any technical data controlled under this subchapter, whether in the U.S. or abroad." 22 C.F.R. § 121.1 (1994). Encryption systems, software, and algorithms are included as "defense articles."

b. Summary of pending federal legislation.

(1) "Encrypted Communications Privacy Act of 1996" (S. 1587). The Senate Bill contains a general declaration that the use of encryption by a U.S. person, domestically or abroad, regardless of the algorithm selected, with or without a key escrow function, and with or without a third-party key escrow holder, is lawful. The Bill would establish limited means by which authorized investigative and law enforcement officials could obtain access to a decryption key. It would also provide for criminal penalties and civil liability for a key holder (escrow agent) who released the key other than either with the consent of the key owner or to authorized investigative or law enforcement officers. Another provision would make all sales of encryption within the U.S. legal, no matter how strong the technology.

(2) "Security and Freedom through Encryption Act" (H.R. 3011). The House Bill declares the use and sale of encryption lawful, except when "in furtherance of a criminal offense" and prohibits the mandatory escrow of a decryption key. In addition, the Bill bars compulsory access to encrypted information by investigative or law enforcement officers, except when such access is obtained pursuant to preexisting law. The Bill would place export control over encryption exclusively in the hands of the Secretary of Commerce.

c. The "clipper chip" debate.

The clipper chip initiative, which originated with the Bush administration and was subsequently endorsed by the Clinton administration, is currently stalled after having met with ferocious opposition. The proposal involved the widespread use of a microprocessor chip that would encrypt and decrypt electronic messages using a public key encryption algorithm which would remain classified. The clipper chip would make it easier for encryption to become standard and enable government eavesdropping on encrypted messages in appropriate circumstances. The eavesdropping aspect of the initiative was to be accomplished by requiring a copy of the private key to be placed in escrow with a government agent, who would release the key only at the request of the owner or pursuant to a judicial order. Opposition to the initiative came from privacy advocates who feared government abuse and from the software and computer industries which voiced concerns that international customers would not purchase products or systems to which the United States government held a key.

C. Managerial.

1. Security audits.

2. Corporate information technology policies.

3. Corporate education advisories.

VI. YEAR 2000 CONCERNS

A. Technical Considerations.

Described by some as a "legal virus," the "Year 2000 Issue" or "The Millennium Crisis" stems from the fact that at multiple levels, computer systems in business, government, and academe have been programmed to recognize dates in two-digit MM/DD/YY format rather than four-digit MM/DD/YYYY format. This format was adopted to save on data storage costs and, possibly, to avoid additional time and expense associated with additional keystrokes. The effect of this oversight, however, is that as the year 2000 approaches, computer systems, particularly legacy systems, will produce inaccurate results or may abort processing altogether. Dates are pervasive in computer systems, being critical to accounting, billing, inventory, health, loan, and personnel record processing. The cost to address the Year 2000 Issue globally has been estimated at $600 billion. According to other estimates, Fortune 50 companies each will have to spend between 35 and 50 cents per line of code to update their existing systems -- a price tag of between $50 to $100 million per company. See generally Peter de Jaeger, Doomsday, ComputerWorld, Sept. 6, 1993, reproduced at: http://www.year2000.com/sw-article.html; Peter de Jaeger, Believe Me, It's Real!, http://www.year2000.com/ believeme.html; and Information Technology Association of America, What Are You Waiting For? (1996).

B. Legal Considerations.

1. Contract law considerations.

a. Rights and responsibilities under existing software/system contracts.

Systems contracts, software licenses, or maintenance agreements may warrant that the system or software will conform to the published specifications of the most recent version. Existing contracts should be reviewed to determine to what extent, if any, Year 2000 problems may be addressed. Contracts now being negotiated should expressly include language addressing the Year 2000 matter.

b. Third-party solutions.

A number of third-party providers offer products and services designed to locate and address potential Year 2000 problems. In exploring the possibility of third-party solutions, counsel should be sensitive to contractual provisions limiting or otherwise affecting the provision of maintenance services by third party providers. For example, maintenance by third-parties may be expressly precluded or may result in voiding the provider's warranty obligations.

2. Copyright law considerations.

a. Third-party maintenance issues.

The U.S. Copyright Act, 17 U.S.C. § 101 and following, affords the computer program copyright owner the exclusive rights to make copies of the work or to prepare adaptations and derivatives. Although 17 U.S.C. § 117 allows *owners* of computer program copies to make additional copies in order to use the program or to ensure that the program will run in a particular hardware/software environment, these statutory rights do not extend to program *licensees*, and computer programs are typically licensed rather than sold. Note, too, that under the "work-for-hire" rules, copyright in custom-developed (by independent third parties) programs inheres in the developer rather than the customer, absent a written assignment of copyright from the developer to the customer and registration of the assignment with the U.S. Copyright Office. Copyright ownership in programs developed internally by an organization's employees vests automatically in the employer. 17 U.S.C. §§ 101 and 201.

The confluence of copyright law and third-party maintenance has been the subject of litigation in several cases, including Triad Systems Corporation v. Southeastern Express Co., 64 F.3d 1330 (9th Cir. 1995), cert. denied, 116 S. Ct. 1015 (1996) and MAI Sys. Corp. v. Peake Computer, Inc., 991 F.2d 511 (9th Cir. 1993). In these cases it was found that the loading of a program into a computer's random access memory (RAM) was a sufficient "fixation" and copying of the work, such that absent authorization of program copyright owner, the program copying performed incident to the provision of hardware maintenance services by a third-party was found to be infringing. Legislation to address this issue is now pending in the U.S. Congress (H.R. 1861).

VII. APPENDICES

A. Computer Fraud and Abuse Act (18 U.S.C. § 1030).

B. Virginia Computer Crime Statute (Va Code Ann. § 18.2-152.1).

C. Virginia Trade Secret Act (Va Code Ann. § 59.1-336 to -343).

D. Listing of State Computer Crime Statutes.

E. Corporate Internet Policy Checklist.

Errata to Written Materials on
Electronic Data: Privacy, Security and Confidentiality

1. Disregard the first sentence of the last paragraph on page IIA-9. For additional information on penalties <u>see</u> Appendix B on pages IIA41-44 and "The Virginia Computer Crime Law" materials by Judge Leslie M. Alden on pages IIB1-9. The statement concerning civil relief refers to Article 7.1 of the Virginia Code (the "Virginia Computer Crimes Act").

2. The last sentence on page IIA-22 should read, "The Virginia Supreme Court affirmed but noted that when the durafoam process ceases to be a trade secret, defendant may petition the Court to modify and limit the injunction."

§ 1030. **Fraud and related activity in connection with computers**

(a) Whoever—

(1) knowingly accesses a computer without authorization or exceeds authorized access, and by means of such conduct obtains information that has been determined by the United States Government pursuant to an Executive order or statute to require protection against unauthorized disclosure for reasons of national defense or foreign relations, or any restricted data, as defined in paragraph y of section 11 of the Atomic Energy Act of 1954, with the intent or reason to believe that such information so obtained is to be used to the injury of the United States, or to the advantage of any foreign nation;

(2) intentionally accesses a computer without authorization or exceeds authorized access, and thereby obtains information contained in a financial record of a financial institution, or of a card issuer as defined in section 1602(n) of title 15, or contained in a file of a consumer reporting agency on a consumer, as such terms are defined in the Fair Credit Reporting Act (15 U.S.C. 1681 et seq.);

(3) intentionally, without authorization to access any computer of a department or agency of the United States, accesses such a computer of that department or agency that is exclusively for the use of the Government of the United States or, in the case of a computer not exclusively for such use, is used by or for the Government of the United States and such conduct adversely affects the use of the Government's operation of such computer;

(4) knowingly and with intent to defraud, accesses a Federal interest computer without authorization, or exceeds authorized access, and by means of such conduct furthers the intended fraud and obtains anything of value, unless the object of the fraud and the thing obtained consists only of the use of the computer;

(5)(A) through means of a computer used in interstate commerce or communications, knowingly causes the transmission of a program, information, code, or command to a computer or computer system if—

(i) the person causing the transmission intends that such transmission will—

(I) damage, or cause damage to, a computer, computer system, network, information, data, or program; or

(II) withhold or deny, or cause the withholding or denial, of the use of a computer, computer services, system or network, information, data or program; and

(ii) the transmission of the harmful component of the program, information, code, or command—

(I) occurred without the authorization of the persons or entities who own or are responsible for the computer system receiving the program, information, code, or command; and

(II)(aa) causes loss or damage to one or more other persons of value aggregating $1,000 or more during any 1-year period; or

(bb) modifies or impairs, or potentially modifies or impairs, the medical examination, medical diagnosis, medical treatment, or medical care of one or more individuals; or

(B) through means of a computer used in interstate commerce or communication, knowingly causes the transmission of a program, information, code, or command to a computer or computer system—

(i) with reckless disregard of a substantial and unjustifiable risk that the transmission will—

(I) damage, or cause damage to, a computer, computer system, network, information, data or program; or

(II) withhold or deny or cause the withholding or denial of the use of a computer, computer services, system, network, information, data or program; and

(ii) if the transmission of the harmful component of the program, information, code, or command—

(I) occurred without the authorization of the persons or entities who own or are responsible for the computer system receiving the program, information, code, or command; and

(II)(aa) causes loss or damage to one or more other persons of a value aggregating $1,000 or more during any 1-year period; or

(bb) modifies or impairs, or potentially modifies or impairs, the medical examination, medical diagnosis, medical treatment, or medical care of one or more individuals;

(6) knowingly and with intent to defraud traffics (as defined in section 1029) in any password or similar information through which a computer may be accessed without authorization, if—

(A) such trafficking affects interstate or foreign commerce; or

(B) such computer is used by or for the Government of the United States;

shall be punished as provided in subsection (c) of this section.

(b) Whoever attempts to commit an offense under subsection (a) of this section shall be punished as provided in subsection (c) of this section.

(c) The punishment for an offense under subsection (a) or (b) of this section is—

(1)(A) a fine under this title or imprisonment for not more than ten years, or both, in the case of an offense under subsection (a)(1) of this section which does not occur after a conviction for another offense under such subsection, or an attempt to commit an offense punishable under this subparagraph; and

(B) a fine under this title or imprisonment for not more than twenty years, or both, in the case of an offense under subsection (a)(1) of this section which occurs after a conviction for another offense under such subsection, or an attempt to commit an offense punishable under this subparagraph; and

(2)(A) a fine under this title or imprisonment for not more than one year, or both, in the case of an offense under subsection (a)(2), (a)(3) or (a)(6) of this section which does not occur after a conviction for another offense under such subsection, or an attempt to commit an offense punishable under this subparagraph; and

(B) a fine under this title or imprisonment for not more than ten years, or both, in the case of an offense under subsection (a)(2), (a)(3) or (a)(6) of this section which occurs after a conviction for another offense under such subsection, or an attempt to commit an offense punishable under this subparagraph;

(3)(A) a fine under this title or imprisonment for not more than five years, or both, in the case of an offense under subsection (a)(4) or (a)(5)(A) of this section which does not occur after a conviction for another offense under such subsection, or an attempt to commit an offense punishable under this subparagraph; and

(B) a fine under this title or imprisonment for not more than ten years, or both, in the case of an offense under subsection (a)(4) or (a)(5) of this section which occurs after a conviction for another offense under such subsection, or an attempt to commit an offense punishable under this subparagraph; and

(4) a fine under this title or imprisonment for not more than 1 year, or both, in the case of an offense under subsection (a)(5)(B).

(d) The United States Secret Service shall, in addition to any other agency having such authority, have the authority to investigate offenses under this section. Such authority of the United States Secret Service shall be exercised in accordance with an agreement which shall be entered into by the Secretary of the Treasury and the Attorney General.

(e) As used in this section—

(1) the term "computer" means an electronic, magnetic, optical, electrochemical, or other high speed data processing device performing logical, arithmetic, or storage functions, and includes any data storage facility or communications facility directly related to or operating in conjunction with such device, but such term does not include an automated typewriter or typesetter, a portable hand held calculator, or other similar device;

(2) the term "Federal interest computer" means a computer—

(A) exclusively for the use of a financial institution or the United States Government, or, in the case of a computer not exclusively for such use, used by or for a financial institution or the United States Government and the conduct constituting the offense affects the use of the financial institution's operation or the Government's operation of such computer; or

(B) which is one of two or more computers used in committing the offense, not all of which are located in the same State;

(3) the term "State" includes the District of Columbia, the Commonwealth of Puerto Rico, and any other commonwealth, possession or territory of the United States;

(4) the term "financial institution" means—

(A) an institution with deposits insured by the Federal Deposit Insurance Corporation;

(B) the Federal Reserve or a member of the Federal Reserve including any Federal Reserve Bank;

(C) a credit union with accounts insured by the National Credit Union Administration;

(D) a member of the Federal home loan bank system and any home loan bank;

(E) any institution of the Farm Credit System under the Farm Credit Act of 1971;

(F) a broker-dealer registered with the Securities and Exchange Commission pursuant to section 15 of the Securities Exchange Act of 1934;

(G) the Securities Investor Protection Corporation;

(H) a branch or agency of a foreign bank (as such terms are defined in paragraphs (1) and (3) of section 1(b) of the International Banking Act of 1978); and

(I) an organization operating under section 25 or section 25(a) of the Federal Reserve Act.

(5) the term "financial record" means information derived from any record held by a financial institution pertaining to a customer's relationship with the financial institution;

(6) the term "exceeds authorized access" means to access a computer with authorization and to use such access to obtain or alter information in the computer that the accesser is not entitled so to obtain or alter; and

(7) the term "department of the United States" means the legislative or judicial branch of the Government or one of the executive departments enumerated in section 101 of title 5.

(f) This section does not prohibit any lawfully authorized investigative, protective, or intelligence activity of a law enforcement agency of the United States, a State, or a political subdivision of a State, or of an intelligence agency of the United States.

(g) Any person who suffers damage or loss by reason of a violation of the section, other than a violation of subsection (a)(5)(B), may maintain a civil action against the violator to obtain compensatory damages and injunctive relief or other equitable relief. Damages for violations of any subsection other than subsection (a)(5)(A)(ii)(II)(bb) or (a)(5)(B)(ii)(II)(bb) are limited to economic damages. No action may be brought under this subsection unless such action is begun within 2 years of the date of the act complained of or the date of the discovery of the damage.

(h) The Attorney General and the Secretary of the Treasury shall report to the Congress annually, during the first 3 years following the date of the enactment of this subsection, concerning investigations and prosecutions under section 1030(a)(5) of title 18. United States Code.[1]

APPENDIX B

§ **18.2-152.1. Short title.** — This article shall be known and may be cited as the "Virginia Computer Crimes Act." (1984, c. 751.)

§ **18.2-152.2. Definitions.** — For purposes of this article:

"Computer" means an electronic, magnetic, optical, hydraulic or organic device or group of devices which, pursuant to a computer program, to human instruction, or to permanent instructions contained in the device or group of devices, can automatically perform computer operations with or on computer data and can communicate the results to another computer or to a person. The term "computer" includes any connected or directly related device, equipment, or facility which enables the computer to store, retrieve or communicate computer programs, computer data or the results of computer operations to or from a person, another computer or another device.

"Computer data" means any representation of information, knowledge, facts, concepts, or instructions which is being prepared or has been prepared and is intended to be processed, is being processed, or has been processed in a computer or computer network. "Computer data" may be in any form, whether readable only by a computer or only by a human or by either, including, but not limited to, computer printouts, magnetic storage media, punched cards, or stored internally in the memory of the computer.

"Computer network" means a set of related, remotely connected devices and any communications facilities including more than one computer with the capability to transmit data among them through the communications facilities.

"Computer operation" means arithmetic, logical, monitoring, storage or retrieval functions and any combination thereof, and includes, but is not limited to, communication with, storage of data to, or retrieval of data from any device or human hand manipulation of electronic or magnetic impulses. A "computer operation" for a particular computer may also be any function for which that computer was generally designed.

"Computer program" means an ordered set of data representing coded instructions or statements that, when executed by a computer, causes the computer to perform one or more computer operations.

"Computer services" includes computer time or services or data processing services or information or data stored in connection therewith.

"Computer software" means a set of computer programs, procedures and associated documentation concerned with computer data or with the operation of a computer, computer program, or computer network.

"Financial instrument" includes, but is not limited to, any check, draft, warrant, money order, note, certificate of deposit, letter of credit, bill of exchange, credit or debit card, transaction authorization mechanism, marketable security, or any computerized representation thereof.

"Owner" means an owner or lessee of a computer or a computer network or an owner, lessee, or licensee of computer data, computer programs, or computer software.

"Person" shall include any individual, partnership, association, corporation or joint venture.

"Property" shall include:

1. Real property;
2. Computers and computer networks;
3. Financial instruments, computer data, computer programs, computer software and all other personal property regardless of whether they are:

a. Tangible or intangible;

b. In a format readable by humans or by a computer;

c. In transit between computers or within a computer network or between any devices which comprise a computer; or

d. Located on any paper or in any device on which it is stored by a computer or by a human; and

4. Computer services.

A person *"uses"* a computer or computer network when he:

1. Attempts to cause or causes a computer or computer network to perform or to stop performing computer operations;
2. Attempts to cause or causes the withholding or denial of the use of a computer, computer network, computer program, computer data or computer software to another user; or
3. Attempts to cause or causes another person to put false information into a computer.

A person is *"without authority"* when he has no right or permission of the owner to use a computer, or, he uses a computer in a manner exceeding such right or permission. (1984, c. 751.)

§ **18.2-152.3. Computer fraud.** — Any person who uses a computer or computer network without authority and with the intent to:

1. Obtain property or services by false pretenses;
2. Embezzle or commit larceny; or
3. Convert the property of another shall be guilty of the crime of computer fraud. If the value of the property or services obtained is $200 or more, the crime of computer fraud shall be punishable as a Class 5 felony. Where the value of the property or services obtained is less than $200, the crime of computer fraud shall be punishable as a Class 1 misdemeanor. (1984, c. 751; 1985, c. 322.)

§ **18.2-152.4. Computer trespass.** — Any person who uses a computer or computer network without authority and with the intent to:

1. Temporarily or permanently remove computer data, computer programs or computer software from a computer or computer network;
2. Cause a computer to malfunction regardless of how long the malfunction persists;
3. Alter or erase any computer data, computer programs or computer software;
4. Effect the creation or alteration of a financial instrument or of an electronic transfer of funds;
5. Cause physical injury to the property of another; or
6. Make or cause to be made an unauthorized copy, in any form, including, but not limited to, any printed or electronic form of computer data, computer programs or computer software residing in, communicated by or produced by a computer or computer network shall be guilty of the crime of computer trespass, which shall be punishable as a Class 1 misdemeanor. (1984, c. 751; 1985, c. 322.)

§ **18.2-152.5. Computer invasion of privacy.** — A. A person is guilty of the crime of computer invasion of privacy when he uses a computer or computer network and intentionally examines without authority any employment, salary, credit or any other financial or personal information relating to any other person. "Examination" under this section requires the offender to review the information relating to any other person after the time at which the offender knows or should know that he is without authority to view the information displayed.

B. The crime of computer invasion of privacy shall be punishable as a Class 3 misdemeanor. (1984, c. 751; 1985, c. 398.)

§ **18.2-152.6. Theft of computer services.** — Any person who willfully uses a computer or computer network, with intent to obtain computer services without authority, shall be guilty of the crime of theft of computer services, which shall be punishable as a Class 1 misdemeanor. (1984, c. 751; 1985, c. 322.)

§ **18.2-152.7. Personal trespass by computer.** — A. A person is guilty of the crime of personal trespass by computer when he uses a computer or computer network without authority and with the intent to cause physical injury to an individual.

B. If committed maliciously, the crime of personal trespass by computer shall be punishable as a Class 3 felony. If such act be done unlawfully but not maliciously, the crime of personal trespass by computer shall be punishable as a Class 1 misdemeanor. (1984, c. 751; 1985, c. 322.)

§ **18.2-152.8. Property capable of embezzlement.** — For purposes of § 18.2-111, personal property subject to embezzlement shall include:

1. Computers and computer networks;

2. Financial instruments, computer data, computer programs, computer software and all other personal property regardless of whether they are:

a. Tangible or intangible;

b. In a format readable by humans or by a computer;

c. In transit between computers or within a computer network or between any devices which comprise a computer; or

d. Located on any paper or in any device on which it is stored by a computer or by a human; and

3. Computer services. (1984, c. 751.)

§ **18.2-152.9. Limitation of prosecution.** — Notwithstanding the provisions of § 19.2-8, prosecution of a crime which is punishable as a misdemeanor pursuant to this article must be commenced before the earlier of (i) five years after the commission of the last act in the course of conduct constituting a violation of this article or (ii) one year after the existence of the illegal act and the identity of the offender are discovered by the Commonwealth, by the owner, or by anyone else who is damaged by such violation. (1984, c. 751.)

§ **18.2-152.10. Venue for prosecution.** — For the purpose of venue under this article, any violation of this article shall be considered to have been committed in any county or city:

1. In which any act was performed in furtherance of any course of conduct which violated this article;

2. In which the owner has his principal place of business in the Commonwealth;

3. In which any offender had control or possession of any proceeds of the violation or of any books, records, documents, property, financial instrument, computer software, computer program, computer data, or other material or objects which were used in furtherance of the violation;

4. From which, to which, or through which any access to a computer or computer network was made whether by wires, electromagnetic waves, microwaves, or any other means of communication;

5. In which the offender resides; or

6. In which any computer which is an object or an instrument of the violation is located at the time of the alleged offense. (1984, c. 751; 1985, c. 322.)

§ **18.2-152.11. Article not exclusive.** — The provisions of this article shall not be construed to preclude the applicability of any other provision of the criminal law of this Commonwealth which presently applies or may in the future apply to any transaction or course of conduct which violates this article, unless such provision is clearly inconsistent with the terms of this article. (1984, c. 751.)

§ **18.2-152.12. Civil relief; damages.** — A. Any person whose property or person is injured by reason of a violation of any provision of this article may sue therefor and recover for any damages sustained, and the costs of suit. Without limiting the generality of the term, "damages" shall include loss of profits.

B. At the request of any party to an action brought pursuant to this section, the court may, in its discretion, conduct all legal proceedings in such a way as to protect the secrecy and security of the computer, computer network, computer data, computer program and computer software involved in order to prevent possible recurrence of the same or a similar act by another person and to protect any trade secrets of any party.

C. The provisions of this article shall not be construed to limit any person's right to pursue any additional civil remedy otherwise allowed by law.

D. A civil action under this section must be commenced before expiration of the time period prescribed in § 8.01-40.1. (1984, c. 751; 1985, c. 92.)

§ **18.2-152.13. Severability.** — If any provision or clause of this article or application thereof to any person or circumstances is held to be invalid, such invalidity shall not affect other provisions or applications of this article which can be given effect without the invalid provision or application, and to this end the provisions of this article are declared to be severable. (1984, c. 751.)

§ **18.2-152.14. Computer as instrument of forgery.** — The creation, alteration, or deletion of any computer data contained in any computer or computer network, which if done on a tangible document or instrument would constitute forgery under Article 1 (§ 18.2-168 et seq.) of Chapter 6 of this Title, will also be deemed to be forgery. The absence of a tangible writing directly created or altered by the offender shall not be a defense to any crime set forth in Article 1 (§ 18.2-168 et seq.) of Chapter 6 of this Title if a creation, alteration, or deletion of computer data was involved in lieu of a tangible document or instrument. (1984, c. 751; 1985, c. 322.)

§ **18.2-152.4. Computer trespass; penalty.** — Any person who uses a computer or computer network without authority and with the intent to:

1. Temporarily or permanently remove computer data, computer programs, or computer software from a computer or computer network;
2. Cause a computer to malfunction regardless of how long the malfunction persists;
3. Alter or erase any computer data, computer programs, or computer software;
4. Effect the creation or alteration of a financial instrument or of an electronic transfer of funds;
5. Cause physical injury to the property of another; or
6. Make or cause to be made an unauthorized copy, in any form, including, but not limited to, any printed or electronic form of computer data, computer programs, or computer software residing in, communicated by, or produced by a computer or computer network shall be guilty of the crime of computer trespass, which shall be punishable as a Class 1 misdemeanor. If such act is done maliciously and the value of the property damaged is $2,500 or more, the offense shall be punishable as a Class 6 felony. (1984, c. 751; 1985, c. 322; 1990, c. 663.)

The 1990 amendment added the last sentence of subdivision 6.

APPENDIX C

CHAPTER 26.

Uniform Trade Secrets Act.

§ **59.1-336. Short title and definitions.** — As used in this chapter, which may be cited as the Uniform Trade Secrets Act, unless the context requires otherwise:

"Improper means" includes theft, bribery, misrepresentation, breach of a duty or inducement of a breach of a duty to maintain secrecy, or espionage through electronic or other means.

"Misappropriation" means:

1. Acquisition of a trade secret of another by a person who knows or has reason to know that the trade secret was acquired by improper means; or

2. Disclosure or use of a trade secret of another without express or implied consent by a person who

a. Used improper means to acquire knowledge of the trade secret; or

b. At the time of disclosure or use, knew or had reason to know that his knowledge of the trade secret was

(1) Derived from or through a person who had utilized improper means to acquire it;

(2) Acquired under circumstances giving rise to a duty to maintain its secrecy or limit its use;

(3) Derived from or through a person who owed a duty to the person seeking relief to maintain its secrecy or limit its use; or

(4) Acquired by accident or mistake.

"Person" means a natural person, corporation, business trust, estate, trust, partnership, association, joint venture, government, governmental subdivision or agency, or any other legal or commercial entity.

"Trade secret" means information, including but not limited to, a formula, pattern, compilation, program, device, method, technique, or process, that:

1. Derives independent economic value, actual or potential, from not being generally known to, and not being readily ascertainable by proper means by, other persons who can obtain economic value from its disclosure or use, and

2. Is the subject of efforts that are reasonable under the circumstances to maintain its secrecy. (1986, c. 210.)

§ **59.1-337. Injunctive relief.** — A. Actual or threatened misappropriation may be enjoined. Upon application to the court, an injunction shall be terminated when the trade secret has ceased to exist, but the injunction may be continued for an additional reasonable period of time in order to eliminate commercial advantage that otherwise would be derived from the misappropriation.

B. In exceptional circumstances, an injunction may condition future use upon payment of a reasonable royalty for no longer than the period of time for which use could have been prohibited. Exceptional circumstances include, but are not limited to, a material and prejudicial change of position prior to acquiring knowledge or reason to know of misappropriation that renders a prohibitive injunction inequitable.

C. In appropriate circumstances, affirmative acts to protect a trade secret may be compelled by court order. (1986, c. 210.)

§ **59.1-338. Damages.** — A. Except where the user of a misappropriated trade secret has made a material and prejudicial change in his position prior to having either knowledge or reason to know of the misappropriation and the court determines that a monetary recovery would be inequitable, a complainant is entitled to recover damages for misappropriation. Damages can include both the actual loss caused by misappropriation and the unjust enrichment caused by misappropriation that is not taken into account in computing actual loss. If a complainant is unable to prove a greater amount of damages by other methods of measurement, the damages caused by misappropriation can be measured exclusively by imposition of liability for a reasonable royalty for a misappropriator's unauthorized disclosure or use of a trade secret.

B. If willful and malicious misappropriation exists, the court may award punitive damages in an amount not exceeding twice any award made under subsection A of this section, or $350,000 whichever amount is less. (1986, c. 210; 1990, c. 344.)

§ **59.1-338.1. Attorneys' fees.** — If the court determines that (i) a claim of misappropriation is made in bad faith, or (ii) willful and malicious misappropriation exists, the court may award reasonable attorneys' fees to the prevailing party. (1990, c. 344.)

§ **59.1-339. Preservation of secrecy.** — In an action under this chapter, a court shall preserve the secrecy of an alleged trade secret by reasonable means, which may include:

1. Granting protective orders in connection with discovery proceedings;
2. Holding in-camera hearings;
3. Sealing the records of the action; and
4. Ordering any person involved in the litigation not to disclose an alleged trade secret without prior court approval. (1986, c. 210.)

§ **59.1-340. Statute of limitations.** — An action for misappropriation shall be brought within three years after the misappropriation is discovered or by the exercise of reasonable diligence should have been discovered. For the purposes of this section, a continuing misappropriation constitutes a single claim. (1986, c. 210.)

§ **59.1-341. Effect on other law.** — A. Except as provided in subsection B of this section, this chapter displaces conflicting tort, restitutionary, and other law of this Commonwealth providing civil remedies for misappropriation of a trade secret.

B. This chapter does not affect:

1. Contractual remedies whether or not based upon misappropriation of a trade secret; or
2. Other civil remedies that are not based upon misappropriation of a trade secret; or
3. Criminal remedies, whether or not based upon misappropriation of a trade secret. (1986, c. 210.)

§ **59.1-342:** Not set out.

§ **59.1-343. Time of taking effect.** — This chapter shall become effective on July 1, 1986, and shall not apply to misappropriation occurring prior to the effective date. With respect to a continuing misappropriation that began prior to the effective date, the chapter also shall not apply to misappropriation that occurs after the effective date. (1986, c. 210.)

APPENDIX D

STATE COMPUTER CRIME STATUS

AL	Computer Crime Act, Code of Alabama, Sections 13A-8-100 to 13A-8-103
AK	Statutes, Sections 11.46.200(a) (3), 11.46.484 (a) (5), 11.46.740, 11.46.985, 11.46.990
AZ	Revised Statutes Annotated, Sections 13-2301(E), 13-2316
CA	Penal Code, Section 502
CO	Revised Statutes, Sections 18-5.5-101, 18-5.5-102
CT	General Statutes, Sections 53a-250 to 53a-261, 52-570b
DE	Code Annotated, Title 11, Sections 931-938
FL	Computer Crimes Act, Florida Statutes Annotated, Sections 815.01 to 815.07
GA	Computer Systems Protection Act, Georgia Codes Annotated, Sections 16-9-90 to 16-9-95
HI	Revised Statutes, Sections 708-890 to 780-896
IA	Statutes, Sections 716A.1 to 716A.16
ID	Code, Title 18, Chapter 22, Sections 18-2201, 18-2202
IL	Annotated Statutes (Criminal Code), Sections 15-1, 16-9
IN	Code, Sections 35-43-1-4, 35-43-2-3
KS	Statutes Annotated, Section 21-3755
KY	Revised Statutes, Sections 434.840 to 434.860
LA	Revised Statutes, Title 14, Subpart D. Computer Related Crimes, Sections 73.1 to 73.5
ME	Revised Statutes Annotated, Chapter 15, Title 17-A, Section 357
MD	Annotated Code, Article 27, Sections 45A and 146
MA	General Laws, Chapter 266, Section 30
MI	Statutes Annotated, Section 28.529(1) - (7)
MN	Statutes (Criminal Code), Sections 609.87 to 609.89
MO	Revised Statutes, Sections 569.093 to 569.099
MS	Code Annotated, Sections 97-45-1 to 97-45-13
MT	Code Annotated, Sections 45-2-101, 45-6-310, 45-6-311
NE	Revised Statutes, Article 13(p) Computers, Sections 28-1343 to 28-1348
NV	Revised Statutes, Sections 205.473 to 205.477
NH	Revised Statutes Annotated, Sections 638.16 to 638.19
NJ	Statutes, Title 2C, Chapter 20, Sections 2c:20-1, 2c:20-23 to 2c:20-34, and Title 2A, Sections 2A:38A-1 to 2A:38A-3
NM	Statutes Annotated, Criminal Offenses, Computer Crimes Act, Sections 30-16A-1 to 30-16A-4
NY	Penal Law, Sections 155.00, 156.00 to 156.50, 165.15 subdiv. 10, 170.00, 175.00
NC	General Statutes, Sections 14-453 to 14-457

ND	Century Code, Sections 12.1-06.1-01 subsection 3, 12.1-06.1-08
OH	Revised Code Annotated, Sections 2901.01, 2913.01, 2913.04, 2913.81
OK	Computer Crimes Act, Oklahoma Session Laws, Title 21, Sections 1951-1956
OR	Revised Statutes, Sections 164.125, 164.377
PA	Consolidated Statutes Annotated, Section 3933
RI	General Laws (Criminal Offenses), Sections 11-52-1 to 11-52-5
SC	Code of Laws, Sections 16-16-40
SD	Codified Laws, Sections 43-43B-1 to 43-43B-8
TN	Code Annotated, Computer Crimes Act, Sections 39-3-1401 to 39-3-1406
TX	Codes Annotated, Title 7, Chapter 33, Section 33.01 to 33.05
UT	Computer Fraud Act, Utah Code Annotated, Sections 76-6-701 to 76-6-704
VA	Computer Crimes Act, Code of Virginia, Sections 18.2-152.1 to 18.2-152.14
WA	Revised Code Annotated, Sections 9A.48.100, 9A.52.010, 9A.52.110 to 9A.52.130
WI	Statutes Annotated, Section 943.70
WY	Statutes, Sections 6-3-501 to 6-3-505

APPENDIX E

Corporate Internet Policy Checklist

1. Introduction
    1.1. Purpose of the Internet Policy
    1.2. Monitoring of Internet Usage
    1.3. Personal Use of the Internet
    1.4. On-Line Etiquette ("Netiquette")
2. Security
    2.1. Authorized Users
    2.2. Passwordsy
    2.3. Divulging Passwords
3. Electronic Mail
    3.1. Introduction
    3.2. E-mail Security
    3.3. Monitoring of E-mail
    3.4. Record Keeping for E-mail Messages
    3.5. General E-mail Etiquette
4. Downloading Information
    4.1. Software
    4.2. Downloading Data - Text, Images, Sound, Video, etc.
5. Uploading Data or Software
6. Fee Based Services
7. Representing the Company

JOHN G. KRUCHKO •#±
JAY R. FRIES •+
PAUL M. LUSKY •
STEVEN W. RAY +#
KATHLEEN A. TALTY •
EDWARD LEE ISLER + #
SUSAN TAHERNIA •#+
JOAN E. BOOK •±
JASON M. BRANCIFORTE#

Admitted • MD + VA # DC ± PA

KRUCHKO & FRIES
COUNSELORS AT LAW

Suite 202
7929 Westpark Drive
McLean, Virginia 22102

Telephone: (703) 734-0554
Telecopier: (703) 734-0876

Suite 305
600 Washington Avenue
Baltimore, Maryland 21204
-----
(410) 321-7310

Suite 900
601 Pennsylvania Avenue, N.W.
Washington, D.C. 20004
----
(202) 347-6550

MONITORING YOUR EMPLOYEES: HOW MUCH CAN YOU DO AND WHAT SHOULD YOU DO WHEN YOU UNCOVER WRONGDOING?

Steven W. Ray, Esq.
Kruchko & Fries

This outline is intended to provide a general overview and is not to be construed as legal advice with respect to specific factual situations

MONITORING YOUR EMPLOYEES: HOW MUCH CAN YOU DO AND WHAT SHOULD YOU DO WHEN YOU UNCOVER WRONGDOING?

Steven W. Ray, Esq.
Kruchko & Fries
7929 Westpark Drive
Suite 202
McLean, Virginia 22102

Employers have long been involved in monitoring the workplace performance of their employees. Technological changes in the last twenty years, however, have significantly enhanced an employer's ability to engage in such monitoring. Employers are capable of monitoring an employee's telephone calls, electronic mail, computer keystrokes, time spent on the telephone, and even time spent in the restroom. These changes in technology have given rise to an increase in the tension between an employer's right to monitor employees to maintain security and employee productivity and the rights of employees to privacy, even in the workplace. This outline summarizes the relevant legal landscape and offers some suggestions to employers seeking to implement an employee monitoring program.

I. MONITORING TELEPHONE COMMUNICATIONS

A. Federal Statutory Law

Title III of the Omnibus Crime Control and Safe Street Act of 1968 (hereinafter "Title III") prohibits any person from intercepting, using or disclosing any wire, oral or electronic communication. 18 U.S.C. § 2511(1). The statute defines "wire communication" to mean any communication by the aid of wire, cable, or other like connection, 18 U.S.C. § 2510(1); "oral communication" to mean any oral communication uttered by a person exhibiting an expectation that such communication is not subject to interception, 18 U.S.C. § 2510(2); and "electronic communication" to mean any transfer of signs, signals, writing, images, sounds, data or intelligence of any nature transmitted by a wire, radio, electromagnetic, photoelectronic or photooptical system. 18 U.S.C. § 2510(12). "Intercept" is defined by the statute to mean "the aural or other acquisition of the contents of any wire, electronic, or oral communication through the use of any electronic, mechanical, or other device." 18 U.S.C. § 2510(4). Such a device is "any device or apparatus which can be used to intercept a wire, oral or electronic communication."

Violation of Title III may result in serious consequences for an employer. The statute provides that any person whose wire, oral, or electronic communication is intercepted, disclosed, or intentionally used in violation of the statute may in a civil action recover statutory damages of $100 a day for each day of violation or $10,000, and punitive damages, if appropriate. 18 U.S.C. § 2520. In addition, a person violating the statute may be subject to a criminal fine or imprisonment or both. 18 U.S.C. § 2511(1), § 2511(4).

Title III, however, contains two critical exceptions that are relevant to employers who monitor employee telephone communications. The first of these, commonly referred to as the "business extension exception," requires both that the instrument used to intercept the call be furnished by a communications provider and that the instrument be used in the ordinary course of the employer's business. Specifically, Congress excepted from the definitions of electronic, mechanical, or other device "any telephone or telegraph instrument, equipment or facility, or any component thereof furnished to the subscriber or user by a provider of wire or electronic communications in the ordinary course of its business and being used by the subscriber or user in the ordinary course of its business." 18 U.S.C. § 2510(5)(a). Unfortunately, little legislative history exists to explain Congress' intent in enacting this exception.

The second of these exceptions is the "consent exception." Title III states that "it shall not be unlawful . . . for a person . . . to intercept a wire, oral, or electronic communication where such person is a party to the communication or where one of the parties to the communication has given prior consent to such interception unless such communication is intercepted for the purpose of committing any criminal or tortious act . . . ." 18 U.S.C. § 2511(2)(d). Therefore, the "consent exception" applies as long as just one of the parties to the communication agrees to the interception.

These two exceptions to Title III have been applied, with varying success, in a handful of cases involving employer monitoring or interception of employee telephone communications. The court's decisions in those cases, which have not been entirely consistent, have established some of the parameters that must be observed by an employer seeking to invoke the exceptions.

Where the employee was employed in a complex area involving the employer's quality control, the employer's interception of his telephone conversations were excepted from Title III, particularly where the employer had provided its employees with a separate phone for personal calls and had informed the employees of its practice of monitoring calls. See Simmons v. Southwestern Bell Tel. Co., 452 F. Supp. 392 (W.D. Okla. 1978).

An employer also did not violate Title III when it listened in on an employee's telephone conversation with a competitor with whom the employee had a close friendship. The parties agreed that the call was a business, not a personal, call, and the court found that the monitoring was in the ordinary course of the employee's business in that it was limited in purpose and time and "was not part of a general practice of surreptitious monitoring." Briggs v. American Filter Co., 630 F.2d 414 (5th Cir. 1980).

Where the conversation clearly is a personal call, however, the employer will have great difficulty in showing that the monitoring of the call occurred in the ordinary course of business. In Watkins v. L.M. Berry & Co., 704 F.2d 577 (11th Cir. 1983), the court held that:

> [A] personal call may not be intercepted in the ordinary course of business . . . except to the extent necessary to guard against unauthorized use of the telephone or to determine whether the call is personal or not. In other words, a personal call may be intercepted in the ordinary course of business to determine its nature but never its contents.

The court also rejected the employer's argument that the employee impliedly consented to the interception because she knew of the employee's monitoring policy. The court held that "[c]onsent under Title III is not to be cavalierly implied," and concluded that the employee had consented to the monitoring of her business calls, but not her personal calls.

A similar conclusion was reached recently by another federal appellate court where an employer suspected his employee of theft and monitored her calls. The employer did not find evidence that she had committed the theft in question, but discovered that she had violated another company rule by selling goods at cost to a man with whom she was having an affair. Despite the fact that the employer learned about the infraction by listening to the employee's calls, the court held that the interception was not in the ordinary course of business because it was the interception of a personal call. Deal v. Spears, 980 F.2d 1153 (8th Cir. 1992).

The clearest conclusion to be drawn from these cases is that an employer, under Title III, is obligated to cease listening as soon as it determines that the call is personal, regardless of the contents of the conversation.

B. State Statutory Law

In addition to federal wiretapping laws, almost every state has enacted statutes addressing the interception of telephone communication. Many of these state statutes are patterned after Title III, and include both the business extension exception and the consent exception. See, e.g., Va. Code Ann. § 19.2-61 et seq. Others have sought to expand the protections afforded under Title III[1] by enacting state laws that, among other things, require the consent of all parties to the communication before the consent defense can be asserted. Some of the state statutes are criminal statutes only, and offer no express civil private right of action, although in many of those states private plaintiffs may assert a common law privacy action based on the state policy embodied in the state statute.

For example, under Florida's wiretapping statute, Fla. Stat. Ann. § 934.01 et seq., all parties to a communication must consent to its interception or disclosure in order for the consent defense to be utilized. Royal Health Care Services, Inc. v. Jefferson Pilot Life Ins. Co., 924 F.2d 215, 218 (11th Cir. 1991). California's wiretapping statute also requires the consent of all parties to the communication before an interception is excepted from the statute's proscriptions. See Cal. Penal Code § 632

Consequently, an employer engaging in the monitoring of employee telephone communications must carefully consider, in addition to Title III, the state wiretapping statutes applicable to the employer's places of business.

---

1 States are permitted to expand the protections of Title III and proscribe wiretapping more restrictively, but any state purporting to legalize an action outlawed by Title III would be preempted by that statute.

C. State Common Law Claims

In addition to, or perhaps in lieu of, any state statutory private cause of action that an employee might have for interception of a telephone communication, an employee may also bring a common law action against an employer for invasion of privacy. Generally, most of these common law privacy actions are based on either the theory that the employer negligently or intentionally breached a duty owed to the employee that is established in the state wiretapping statute, or, more commonly, on the premise that the employer has intruded upon the "seclusion" of the employee. Establishing the latter usually requires the individual asserting the privacy claim to show that the defendant committed an intentional intrusion, which a reasonable person would find objectionable or offensive, into the plaintiff's privacy or seclusion.

For example, in Pemberton v. Bethlehem Steel Corp., 66 Md. App. 133, 502 A.2d 1101, cert. denied, 508 A.2d 488 (1986), the court considered an invasion of privacy claim asserted by a union agent who claimed that an employer who employed some of the union's members had him placed under surveillance and thus intruded into his seclusion. The court held that the "gist of the offense is the intrusion into a private place or the invasion of a seclusion that the plaintiff has thrown about his person or affairs. There is no liability for observing him in public places, 'since he is not then in seclusion.'" Even if the employer's surveillance constituted an intrusion, the court further held, the surveillance would only be actionable if the intrusion would be highly offensive to a reasonable person. Thus, it is likely that whether an employer has engaged in a common law invasion of privacy by monitoring employee telephone communications will depend largely upon the employee's ability to show that his communication took place under circumstances that a court would find to be private and in a manner that a reasonable person would consider offensive.

To show that an intrusion was into a private place, a plaintiff alleging this type of common law claim probably must show that he had a reasonable expectation of privacy in the intercepted communication. For example, in Simmons v. Southwestern Bell telephone Co., 452 F. Supp. 392 (W.D. Okl. 1978), where an employee alleged a Fourth Amendment privacy right, the court held that, even had the plaintiff shown that his employer was a state actor so as to implicate the Constitution, he could not establish a reasonable expectation of privacy since the employer had a clearly established and communicated practice of monitoring employee telephone calls for service quality checks. See also Faulkner v. Maryland, 317 Md. 441, 564 A.2d 785 (1989) (holding that employee could not have had reasonable expectation of privacy in a locker because employer had expressly reserved right to inspect lockers).

II. MONITORING EMPLOYEE ELECTRONIC MAIL AND VOICE MAIL

A. Electronic Mail (E-mail)

As networked personal computers have proliferated throughout the business environment in the last ten years, there has been a concomitant expansion in the number of employees who now access some type of electronic mail ("E-mail") system as part of their daily routine. As originally enacted, Title III applied only to wire and oral communications and thus offered no protection to E-mail messages. In 1986, however, the protections of Title III were extended to "electronic communications" by the passage of the Electronic Communications Privacy Act ("ECPA"). One of the principal purposes behind the amendment of Title III was to offer non-aural communications, including E-mail, the same protection as was accorded wire/telephone communications. To achieve this goal, "electronic communication" was broadly defined to mean "any transfer of signs, signals, writing, images, sounds, data, or intelligence of any nature transmitted in whole or in part by a wire, radio, electromagnetic, photoelectronic or photooptical system that affects interstate or foreign commerce." 18 U.S.C. § 2510(12). The legislative history of the ECPA is clear that electronic mail ("E-mail") was intended to be covered as an electronic communication.

By including electronic communication in the same provision as wire or oral communication, however, Congress made the monitoring of electronic communications subject to the same exceptions as had been afforded interception of wire or oral communications. Thus, an employer who chooses to monitor its employee's E-mail messages for business reasons most likely would be protected under the business extension exception. For example, just as employers are permitted to monitor telephone communication to ensure that employees are not spending too much company time engaged in personal calls, <u>see, e.g.</u>, <u>Deal</u>, 980 F.2d at 1158, it would appear that an employer also would have a similar business purpose in monitoring E-mail messages to ensure that employees are not spending too much time exchanging personal messages. Moreover, by advising employees of its intent to periodically monitor E-mails, employers can assert that employees have given implied consent to such monitoring.

Furthermore, the provisions of the ECPA include an additional exception that employers seeking to monitor E-mail should be able to utilize as a defense to such monitoring. Under the ECPA, the provider of electronic communications can access stored communication without running afoul of the Act. Because, in the context of the corporate environment, the employer is the system provider, the employer arguably can access and review stored E-mail messages without violating the ECPA. There appear to be no reported decisions at this time regarding an employer's right to monitor employee E-mail, but several such cases appear to be pending in California, and it is inevitable that other cases will arise in the near future. In the meantime, as long as employers can meet either the requirements that courts have developed for the business extension exception or the consent exception, or can take advantage of the stored communications exception, monitoring of employee E-mail is probably permissible if conducted in a reasonable manner (i.e., no excessive reading of personal E-mail) and if the results of the monitoring are not improperly disclosed.

B. <u>Voice Mail</u>

As with E-mail, in the last ten years, "voice mail" has become prevalent in the business environment, allowing callers the option of leaving a message in an employee's "voice mailbox." The proliferation of voice mail raises additional questions about an employer's right to monitor employee communications in the workplace. Although there appear to be no reported decisions involving an employer's surreptitious interception of an employee's voice mail, at least one case is pending in New York that may address whether voice mail is entitled to the same, less, or greater protection than live telephone communications under either Title III, state statutes, or state common law privacy rights. The result in that case, or in others to follow, likely will be that voice mail is afforded, at a minimum, the same protection as live conversation, and arguably greater protection.

First, although it is unclear that voice mail is covered by Title III, most courts likely will consider voice mail messages to be either wire or electronic communications. If this is the case, then an employer will have to rely upon either the business extension exception or the consent exception in Title III. Second, employees may be able to bring an action under a more restrictive state statute or under a common law right of privacy. The employee may be able to establish the latter based on a showing of a reasonable expectation of privacy. Unlike live conversations, most voice mailboxes may be accessed only through a password or numerical code. As a result, an employee may have a greater expectation of privacy in a voice mailbox than when engaging in live conversation. Consequently, employers should exercise extreme care in accessing employee voice mailboxes.

III. SUBJECTING EMPLOYEES TO POLYGRAPH EXAMINATIONS

A relatively routine method of detecting employee theft or misappropriation would be the use of a polygraph or lie detector test, particularly in those industries, such as the communications industry, where the employee's misappropriation might not become palpable until long after the offense, if at all. Unfortunately for employers, federal and state statutes prohibit random polygraph examinations, and their use, even in furtherance of a specific investigation, must be carefully administered.

A. Federal Law

The federal Employee Polygraph Protection Act of 1988 prohibits employers from requiring employees or applicants for employment to submit to a lie detector test except in very limited circumstances. 29 U.S.C. § 2002(1). Employers also are prohibited from discriminating against, disciplining, or discharging an employee who refuses to take a polygraph. 29 U.S.C. § 2002(3). Violations of the Act can result in the imposition of civil penalties of not more than $10,000 as well as the institution of private actions for equitable relief including reinstatement, promotion, or the payment of lost wages. 29 U.S.C. § 2005.

The Polygraph Act includes several exceptions to its proscriptions, however. The most important of these to private employers is the limited exemption for "ongoing investigations." Utilization of that exemption requires the employer satisfy a number of conditions. Specifically, the employer may require an employee take a polygraph if:

- The test is administered in connection with an ongoing investigation involving economic loss or injury to the employer's business, such as theft, embezzlement, misappropriation, or an act of unlawful industrial espionage or sabotage;

- The employee had access to the property that is the subject of the investigation;

- The employer has a reasonable suspicion that the employee was involved in the incident or activity under investigation; and

- The employer executes a statement, provided to the examinee before the tests, that --

    - sets forth with particularity with specific incident or activity being investigated and the basis for testing particular employees;
    - is signed by a person (other than a polygraph examiner) authorized to legally bind the employer;
    - is retained by the employer for at least three years; and
    - contains at a minimum --
      -- an identification of a specific economic loss or injury to the business of the employer;
      -- a statement indicating that the employee had an access to the property that is a subject of the investigation; and
      -- a statement describing the basis of the employer's reasonable suspicion that the employer was involved in the incident or activity under investigation.

29 U.S.C. § 2006(d). In addition, the Polygraph Act states that an employer may not take action against an employee based on the results of the polygraph unless the employer has additional supporting evidence of the employee's involvement in the alleged offense. 29 U.S.C. § 2007.

The regulations to the Polygraph Act further define ongoing investigation as requiring the investigation of a specific incident or activity. Thus, the regulations explain an employer would not be permitted to subject an employee to a polygraph in an effort to determine whether any theft has, in fact, occurred. Furthermore, the regulations prohibit the use of a polygraph where the employer generally suspects that theft is occurring because of a high-loss of inventory, unless the employer is investigating a specific loss of a specific inventory and has a reasonable suspicion that a particular employee was involved. 29 C.F.R. § 801.12(b). A "reasonable suspicion" is an observable basis in fact, such as information from a co-worker or an employee's behavior or demeanor which indicates a particular employee's involvement, and mere access or opportunity does not give rise to a reasonable suspicion. 29 C.F.R. § 801.12(f).

The impact of the Polygraph Act is the virtual elimination of the polygraph or lie detector, which is broadly defined under the Act, as a means of preliminary investigation of employee misconduct. Only after the employer has developed a reasonable suspicion and has satisfied the requirements for administering a polygraph under the Act may the employee actually be subjected to a polygraph. The employer was held to have such a reasonable suspicion in <u>In re Scrivener Oil Co.</u>, 7 I.E.R. Cas. 962 (1992), where the subject employee was working alone when at the time that the employer developed a large cash shortage. Because the employer complied with the notice requirements of the Polygraph Act, the employee's polygraph was not actionable under the Act. The employer was less fortunate in In re Rapid Robert's Inc., 7 I.E.R. Cas. 946, where the employer failed to satisfy the Act's requirements, even though it had reasonable suspicion to suspect the employee of theft, because it did not provide the employee with sufficient advance notice of the examination.

B. State Laws

Most states also have enacted laws prohibiting employers from subjecting employees to polygraph examinations, some of which are more restrictive than the federal Polygraph Act and provide greater potential remedies to the aggrieved employee. For example, in the District of Columbia, employers are completely prohibited from subjecting employees to polygraph examinations. D.C. Code Ann. § 36-802. No "ongoing investigation" exemption exists under that statute, and violation of the D.C. law "shall be an unwarranted invasion of privacy in the District of Columbia, and shall be compensable by damages for tortious injury." In addition to an amount of damages to be "established by the court," the employer who violates that act may also be liable for attorney's fees and guilty of a misdemeanor. D.C. Code Ann. § 36-803.

Consequently, employers should take care to satisfy the requirements of the applicable state polygraph statute as well as the federal Polygraph Act before administering a polygraph to an employee, even as part of an ongoing investigation.

IV. WORKPLACE SEARCHES AND VIDEO SURVEILLANCE

Despite the number of statutes that have been enacted, at both the federal and state level, prohibiting or restricting an employer's ability to monitor telephone or electronic communications or to subject employees to polygraphs, employers are left relatively unfettered with regard to perhaps the most intrusive forms of employee monitoring, the physical search and video surveillance of a workplace.

A. Workplace Searches

1. Office and Desk Searches

Almost all of the developments in the area of workplace searches have involved public employees who have asserted Constitutional Fourth Amendment rights against being subjected to an unreasonable search and seizure. Because the employee's manager or supervisor generally is considered to be a government actor, such protections are deemed to apply. Prior to 1987, the application of the Fourth Amendment protections in the workplace of a public employer was somewhat inconsistent as courts wrestled with the extent to which a public employee may have a reasonable expectation of privacy in various aspect of the employee's work environment, such as the employee's desk, locker, and even briefcase.

The United States Supreme Court finally considered the issue in O'Connor v. Ortega, 107 S. Ct. 1492 (1987). The Court stated, first, that the workplace includes hallways, cafeteria, offices, desks, and file cabinets, even if an employee places personal items in those places. The Court also noted that "[n]ot everything that passes through the confines of a business address can be considered part of the workplace context, however." Specifically, the Court found that public employees may maintain their expectation of privacy in some items, such as suitcases or purses, even where they are brought into the workplace. The Court thus rejected the argument of the Solicitor General that a public employee can never have a reasonable expectation of privacy in the workplace, finding instead that each employee's expectation of privacy must be assessed in the context of the employment relationship.

> Public employees' expectations of privacy in their offices, desks, and file cabinets, like similar expectations of employees in the private sector, may be reduced by virtue of actual office practices and procedures, or by legitimate regulation.

Emphasis added under the facts of that case, the Court held that an employee who had been subjected to his employer's search had a reasonable expectation of privacy in his desk and file cabinet, because he had maintained the same office for 17 years and did not share it with anyone. The Court then held that whether the employer had violated this expectation of privacy where its search was motivated by investigation of work related misconduct depended upon the reasonableness of the search. The case was remanded to the lower courts for this determination.

Although the O'Connor decision does not have direct application for private employers, its holding undoubtedly will provide guidance to courts that are faced in the future with determining the extent to which private employers may engage in workplace searches. See, e.g., Okura & Co. v. Careau Group, 783 F. Supp. 482, 505-06 (C.D. Cal. 1991)(court rejected invasion of privacy claims filed by corporate board members because it found, citing O'Connor, that the board members did not have a reasonable expectation of privacy in their offices vis-à-vis the CEO of the corporation who conducted the office searches).

Even though Fourth Amendment protections do not extend to private employees as regards their private employers, courts can also be expected to apply the Fourth Amendment analysis to searches by private employers. Employers thus can best protect themselves by communicating to employees the employer's right to conduct reasonable workplace searches of desks and file cabinets, thus reducing the employees' expectation of privacy in those areas. In addition, employers can reduce their exposure to invasion of privacy claims by limiting their searches to occasions where they have a reasonable suspicion of employee wrongdoing. See, e.g., Faulkner v. Maryland, 317 Md. 441, 564 A.2d 785 (1989) (private employer's search of employee's locker with police attending, even if constituting state action, was reasonable in light of employer's well founded belief that drugs and alcohol were being stored in employee lockers and in light of the employer's express reservation of the right to search employee lockers).

2. Physical Searches

Physical searches of an employee's person are almost unheard of in the context of private employers, but it is clear that any such search likely would constitute an invasion of privacy. In Bodewig v. K-Mart, Inc., 54 Ore. App. 480, 635 P.2d 657 (1981), a female employee accused by a female customer of stealing $20 was required to enter a dressing room and, in the presence of a female supervisor and the customer, disrobe down to her underwear. The Court held that the employee stated tort claims for outrageous conduct and infliction of emotional distress arising out of the search.

B. Video Surveillance

There have been surprisingly few reported decisions on the issue of whether an employer's video surveillance constitutes an invasion of employee privacy. In one case, Marrs v. Marriott Corp., 830 F. Supp. 274 (D. Md. 1992), a security supervisor who suspected that someone was looking through the locked drawers of his desk received permission from the employer to install a hidden video camera in the office. The video camera taped a night security guard picking the desk drawer with a paper clip. After the guard was terminated, he sued, claiming, among other things, that the hidden videotaping was an intrusion upon his seclusion and thus an invasion of his privacy. Not unexpectedly, the court held that the employee had no reasonable expectation of privacy in an open office that all of the security guards could access.

A similar result was reached on slightly different grounds In Saldana v. Kelsey-Hayes Co., 443 N.W.2d 382, 4 I.E.R. Cas. 1107 (Mich. Ct. App. 1989). There, an employee was injured after suffering a fall in his workplace. When the employee claimed a work-related disability, his employer hired a private investigative firm to determine whether and to what extent the employee was really injured. The investigative firm observed the employee in public, unbeknownst to the employee, and through the open windows of the employee's home, using both the naked eye and a powerful camera lens. The employee, when he learned of the surveillance, asserted a claim for invasion of privacy. The court held, first, that observation of the employee through an open window with the naked eye would not be considered as intrusive, but that whether the use of the camera lens was intrusive was a jury question. The court then concluded, however, that it was irrelevant whether the use of the lens was intrusive because the intrusion was not into matters that the plaintiff had a right to keep private, given the employer's interest in ensuring that the employee was not engaging in fraud by claiming disability.

Employers may engage in videotaping of the workplace because the employee does not have a reasonable expectation of privacy there, particularly if the employer discloses the presence of the cameras, and because the events that take place in the work environment are of legitimate interest to the employer. Of course, the employer should exercise good judgment and refrain from placing video cameras in places such as employee restrooms where a court almost certainly would conclude that the employees have a reasonable expectation of privacy.

V. PROPOSED LEGISLATION

In the last Congress, Senator Paul Simon (D-Ill.) proposed new legislation, entitled the Privacy for Consumer and Workers Act ("PCWA"), which would require that all electronic monitoring by employers be relevant to the employee's work performance and that employees, customers and the public be given notice of such monitoring.

Under the PCWA, as proposed, the term "electronic monitoring" would mean the "collection, storage, analysis or reporting of information concerning an employee's activities by means of a computer, electronic observation and supervision, telephone service observation, telephone call accounting, or other form of visual, auditory, or computer based technology which is conducted by any method other than direct observation by another person including the following methods: transfer of signs, signals, writing, images, sounds, data, or intelligence of nay nature which are transmitted in whole or in part by a wire, radio, electromagnetic, photoelectronic, or photo-optical system." The bill would permit an employee to have access to data collected about the employee's work performance and would limit disclosure and use of such data by the employer.

Although the bill would permit electronic monitoring, no employer would be permitted to engage in such monitoring in bathrooms, locker rooms or dressing rooms, unless the employer has a reasonable suspicion that an employee is engaged in conduct which violates civil or criminal law. Moreover, lawful electronic monitoring would be restricted to a periodic or random basis and could only be done under the following conditions: (1) for new employees, random or periodic monitoring could occur for up to 60 days; (2) for employees with more than 60 days of tenure but less than 5 years, periodic or random monitoring would be limited to not more than two hours in any week and employees must be given notice of the monitoring at least 24 hours but not more than 72 hours before the monitoring begins; and (3) for employees with more than 5 years tenure, no electronic monitoring would be permitted unless the employer has a reasonable suspicion that the employee is engaged in conduct which violates criminal or civil law or constitutes willful gross misconduct, and this misconduct would adversely affect the employer's interest or the interest of such employer's employees.

An employer who engages in electronic monitoring would be required to post a notice from the Secretary of Labor which would inform employees about their rights under the PCWA. In addition, the employer would be required to provide each employee who would be electronically monitored with prior written notice about the monitoring. The employer's written notice would contain two parts, one part outlining the nature, scope and use of the monitoring[2] and the other part explaining where the employer is not required to give prior notice about monitoring.[3]

---

2 The notice must include the following: (1) the form of electronic monitoring to be used; (2) the personal data to be collected; (3) the hours and days per week that electronic monitoring will occur; (4) the use to be made of personal data collected; (5) interpretation of printouts of statistics or other records of information collected through electronic monitoring if the interpretation affects the employees; (6) existing production standards and work performance expectations; and (7) methods for determining production standards and work performance expectations based on electronic monitoring statistics if the methods affect the employees.

3 The exception to notice requirement applies where an employer has a reasonable suspicion that the employee is engaged in conduct which (1) violates criminal or civil law or constitutes willful gross misconduct, and (2) adversely affects the employer's interest or the interest of such employer's employees.

In addition, employers would be required to provide general notice about electronic monitoring to prospective employees and to give written notice to any prospective employee to whom an employment offer is made. Customers and the public would also be entitled to notification of electronic monitoring if the activity would encompass customers or members of the public.

The PCWA failed to make it out of Senate Labor Committee in the last Congress, although the Committee did take several days of testimony regarding the legislation. A representative from Senator Simon's office stated that reintroduction of the bill in the new Republican-controlled Congress is presently under consideration and, as of February 1995, no decision had been made regarding the bill's future. Because of the impact that legislation such as the PCWA would have on employer monitoring of employees, and because much of the testimony previously taken by the Committee was from representatives of employee interest or individual rights groups, such as the Communications Workers of America and the ACLU (although some employers, such as MCI, also were represented), employers, particularly those in the communications industry, should carefully monitor such legislation in the future.

VI. GUIDELINES FOR MONITORING EMPLOYEES

Employers should establish some guidelines for employee monitoring and have such guidelines reviewed by counsel to ensure compliance with relevant federal and state laws. Because the law may differ slightly from state to state, it is difficult, if not impossible, to draft a uniform policy, but the following are some guidelines that employers should consider in implementing employee monitoring.

A. Determine the Monitoring Necessary for Protection of Business Interests

In many respects, whether defending against a statutory wiretapping claim or a breach of privacy allegation, the linchpin of an employer's defense is its ability to show that the employee monitoring was related to and justified by business necessity. Employers therefore should consider the nature of their business and outline those areas where employee monitoring would be justified. For example, an employer that utilizes telemarketing or consumer service personnel may need to monitor calls to ensure that appropriate customer relations are being observed.

Employers who have reasonable cause to suspect that illegal activities are taking place on the premises may also institute monitoring to eradicate any such activities in the interest of the business. If the illegal activity is an offense against the employer, such as theft or misappropriation of company property, then obviously business interests are implicated. Even if the activity is not against the employer directly, for example, the selling of illegal drugs in the employee locker room, then the employer may engage in reasonable monitoring because it is within an employer's business interests to ensure that crimes are not being perpetrated on the employer's property.

B. Consider the Impact of Applicable Laws

As noted above, state laws differ as to those activities that are considered permissible. For example, some states prohibit polygraph examinations of employees and some state constitutions specifically offer protection of privacy interests and may thus more easily give rise to a common law invasion of privacy claim. Consequently, employers should determine which state law governs their employees and design their employee monitoring system in accordance with that law. For employers with operations in different states, this may require the employer to maintain monitoring techniques that differ depending on the location of the facility where the employees are employed.

C. Communicate the Monitoring Policy to Employees

In addition to establishing the business necessity of monitoring, employers can best protect themselves from wiretapping or invasion of privacy claims by obtaining employee consent to monitoring, even if that consent is simply implied from the fact that the employer made its monitoring practices well known to its employees. Some employers may hesitate to communicate its intent to monitor on the basis that monitoring creates low employee morale and causes friction between employees and management. Although this will always be true to some extent, employers can limit the negative impact of employee monitoring by being straightforward with employees and by explaining that such monitoring is for the protection of the employee as well as the employer. In the end, employers should not sacrifice the need to obtain implied employee consent out of fear that communication of the monitoring policy will damage employer-employee relations.

An employer can communicate its monitoring policy in a number of ways:

- *Employee handbook or manual.* Notify the employees in the handbook that the employer engages in monitoring for business reasons, explain the nature of the monitoring, and state that the employee is presumed to have knowledge that his telephone conversations, E-mail, etc. may be monitored.

- *Other written communication.* Interoffice memoranda or handouts to employees can reiterate the policy contained in the handbook, and annual distribution of such handouts will negate the claim of the employee who asserts that he has not read the employee manual since he commenced employment and that the policy was not in the manual at that time.

- *Posting.* Post the monitoring policy on employee bulletin boards or in employee lounge areas.

- *Signed agreements.* Employers may want to include a communication about employee monitoring in other agreements that the employee is required to sign, such as a confidentiality agreement or a non-compete agreement. The employee's signature will provide the employer with express consent to monitor, provided the monitoring that takes place comports with the monitoring described in the agreement.

- *Employee meetings.* In order to defuse employee anxiety about monitoring and to communicate the policy, the employer may hold meetings with employees where the monitoring is explained and where employees can ask questions. Recording attendance at such meetings is advisable in the event that the employer later seeks to assert the employee's presence at the meeting as evidencing implied consent.

D. Establish Reasonable Limits on Monitoring

An employer's monitoring policy and practices should be tailored to protect its business interests, and should not be overbroad either in design or in implementation. Courts have tended to view unlimited or unfettered monitoring practices with disfavor. Thus, for example, employers generally should not monitor "personal" communications or undertake overly intrusive surveillance measures, such as the placement of video cameras in restrooms or by the entrances to restrooms, in locker rooms, or in employee lounges.

E. Train Managers and Supervisors to Observe Acceptable Limits

In addition to designing reasonable limits on monitoring, employers should take measures to ensure that the limits are honored by the managers or supervisors with access to the information gathered in the monitoring. If an employee learns that a manager entertained himself by reading the employee's E-mail, the employer may face an unnecessary invasion of privacy claim.

F. Maintain Procedures for Use and Disclosure of Monitoring Results

Employers should also design and implement the means by which the results of employee monitoring will be used and disclosed. For example, if the employer randomly tapes telephone conversations between employees and customers, the employer should have procedures in place that specify the individual responsible for screening those tapes, the secure location where the tapes will be maintained, and the period for which the tapes will be stored before being erased.

VII. ACTIONS UPON DISCOVERING WRONGDOING

One of an employer's worst nightmares is to discover that an employee has been engaging in theft, embezzlement, fraud, or some other offense against the company. Unfortunately, such events do occur, and an employer must be prepared to respond when an employee offense is uncovered. Appropriate employer response is important not only so that the employee may be prosecuted for his actions, but also so that the employer may determine, to the greatest degree possible, the full nature and extent of the damage caused to the company by the employee's actions. Although every situation will necessarily vary according to its facts, the following is a suggested checklist of steps the employer should consider taking when it first receives notice of possible wrongdoing by an employee.

A. Act Quickly But Prudently.

Obviously if a crime is being committed against the employer, a rapid response is necessary. Failure to act may result in further harm to the company or in the loss of valuable evidence that is needed to prove the wrongdoing. At the same time, however, employers must balance the need to act quickly with prudence. An employer that acts imprudently or rashly may find that it has taken action against an innocent employee and perhaps has exposed itself to numerous claims by the accused employee, including claims for defamation and intentional infliction of emotional distress. The employee also likely will be sued for discrimination if the employee is a member of a protected class. Employers thus must balance speed with prudence.

B. Notify In-house Counsel and/or Outside Legal Counsel

The legal ramifications of employee malfeasance are significant, both for the employer and the employee. Counsel should be contacted immediately when criminal activity is suspected.

C. Confirm, to the Extent Possible, the Misconduct

Employers will serve themselves well by taking the necessary time to gather sufficient facts and to preserve evidence before acting against an employee. What may seem to be an egregious criminal offense at first glance may turn out to nothing more than a slight infraction of company policies. Confirmation of the misconduct is also important because the employer will need to produce evidence of the wrongdoing if, in fact, criminal misconduct is involved. Gathering and preserving evidence is critical in the early stages of an investigation.

D. Keep Information on a "Need-to-Know" Basis

The investigation and confirmation of the wrongdoing should be conducted using a minimal number of personnel. If employees learn that an investigation is underway, the employer may be further damaged in a couple of respects. First, the employee who is suspected of engaging in the wrongdoing may get wind of the investigation and destroy valuable or necessary evidence of the wrongdoing. As a result the employer may be unable to prove the crime occurred, or may be unable to assess the severity of the criminal activity. Second, by allowing word of an investigation to seep out, the employer may subject itself to a defamation claim when the other employees learn that a co-worker is being investigated, especially if it turns out that he is innocent of any activity rising to the level of criminal conduct. Employers thus should take precautions to maintain the confidentiality of the investigation.

E. Apply Company Policies in a Consistent and Nondiscriminatory Manner

Before acting on an employee's misconduct, be sure that the company is applying its policies in a manner that is consistent with past practice. For example, if an employee has been caught giving a family member access to free long distance service, an action that unquestionably both violates company policy and the law, the employer must also consider how it has treated previous violators, if any have existed. If the accused employee is a member of a protected class (minority, disabled, older worker, etc.), and is treated more harshly than others in the past, the employer likely has exposed itself to civil liability, even if the employee actually cngaged in the misconduct. If, for example, a company has a well established history of merely reprimanding white males who are caught giving family members access to long distance service, and then discharges a black employee who has engaged in the same conduct, the employer will not fare particularly well in the ensuing discrimination action. Employers should apply company policy and practice consistently.

F. Depending on the Offense, Contact Law Enforcement Authorities

For obvious reasons, as soon as evidence of criminal activity is confirmed, law enforcement authorities should contacted.

G. Confront the Employee

Before the employee is summarily terminated and charged with criminal offenses, the employer should (perhaps in the presence of law enforcement officials or in-house security personnel, depending once again on the seriousness of the conduct) confront the employee. At least two witnesses should be present. This will provide the employee with an opportunity to explain the misconduct and present a defense of his actions. If it later turns out that the employee is not guilty of the alleged misconduct, and if the employer failed to give the employee some semblance of due process, an employer might have difficulty presenting its defense in the ensuing civil action for wrongful discharge. For this same reason, confrontation of the employee should take place away from the employee's regular worksite so that the employee, should the charges prove to be false, will not be unduly humiliated in front of co-workers. Such humiliation will undoubtedly give rise to claims for defamation and intentional infliction of emotional distress.

Computer Crime on the Internet – Sources and Methods

- Christine Axsmith

To start, the FBI representative will discuss recent examples of system break-ins, how they were accomplished, and investigations of computer crime cases. A Department of Justice attorney will highlight changes in the area of computer crime law, focusing on evidentiary issues that every systems person should know to preserve evidence in the event of an attempted break-in. The CERT representative will discuss steps to take in the event of an illegal hacker attempt on the system and what resources are available to help should an attempted break in occur. After that, questions will be taken from the floor.

The panel participants will be:

Mark Pollitt, FBI - Mr. Pollitt has a great deal of experience in investigating computer crime, and will elaborate on the lessons learned from that experience.

Phil Reitinger, Department of Justice - Mr. Reitinger has a great deal of experience prosecuting computer crime in the Department of Justice. He will describe the specifics about the recent changes in the Computer Fraud and Abuse Act and give examples on what that means in terms of law enforcement.

Barbara Fraser, Computer Emergency Response Team - Ms. Fraser has experience with the Internet and security issues which will enrich the discussion about decriminalizing certain forms of hacking.

Legal Liability for Information System Security Compliance Failures: New Recipes for Electronic Sachertorte Algorithms

Panel Members, Affiliations and Statements

Fred Chris Smith, Trial Attorney in Private Practice in Santa Fe, New Mexico, Special Prosecutor and Computer Security Consultant

Fred Smith has practiced civil and criminal law in Colorado and New Mexico since graduating from Stanford Law School in 1972. He received a B. A. in philosophy from the University of Michigan. Fred served as the Director of Special Prosecutions and Investigations and as Director of Antitrust Enforcement for four New Mexico Attorneys General. In 1988-9 he served as the first Director of the National Association of Attorneys General RICO Enforcement Project in Washington D.C., which established special financial crime and civil litigation units in Arizona, Colorado, Oregon and Washington. Since 1985, he has developed and presented computer crime training programs for investigators and prosecutors throughout the United States. Since 1993, he has coordinated annual training conferences at Los Alamos National Laboratory in conjunction with SEARCH and the New Mexico High Tech Crime Investigation Association, for law enforcement and corporate security professionals, providing intermediate and advanced training in computer security policies and procedures, and Internet crime detection and prosecution.

John Montjoy, Sr. VP and General Counsel, BBN Corporation

John Montjoy is the Senior Vice President and General Counsel of BBN Corporation, the leading independent provider of Internet services. He graduated from Tulane University Law School in 1969 and joined BBN as General Counsel in 1984. His responsibilities include all legal, regulatory and contractual affairs of the company. Before joining BBN he was in the legal department of Signal Cos., Cincinnati Milacron and Schlumberger Ltd. John has more than 25 years experience in computer law and continues to be very active in the formation of law and in solving legal problems related to the Internet. He was a founder of the Internet Law and Policy Forum, a not-for-profit non-governmental organization composed of approximately thirty leading Internet companies around the world. He currently serves on the executive committee of the Forum.

Edward Tenner, Writer and Visiting Researcher in the Department of Geological and Geophysical Sciences at Princeton University

Edward Tenner was formerly the executive editor for physical science and history at Princeton University Press. In 1995-6 he was a Fellow of the Woodrow Wilson International Center for Scholars served as a consultant to the Jerome and Dorothy

Lemelson Center for the History of Invention and Innovation, National Museum of American History, Smithsonian Institution to select inventors whose work will be documented and studied. He received an A.B, from Princeton and his Ph.D. in history from the University of Chicago. He has held visiting research positions at Rutgers University and the Institute for Advanced Study. In 1996, his book, Why Things Bite Back: Technology and the Revenge of Unintended Consequences, was published by Alfred Knopf.

David J. Loundy, Internet Service Provider Attorney in Private Practice in Highland Park, Illinois, and an Electronic Publisher of His Own Computer Law Related Articles

David Loundy practices law in the Chicago area and provides legal representation to ISPs and other clients with on-line content legal issues. He graduated with distinction from Purdue University with a B.A. in Telecommunications. He received his J.D. with distinction from the University of Iowa College of Law. David has published a number of articles on a wide range of computer related legal topics. He writes a monthly column on *Technology Law* for the Chicago Daily Law Bulletin and a monthly column appearing in The Cyberspace Lawyer entitled, *E-Law*. In 1993 and 1994 he published articles on computer information system law and system operator liability in the E-Law Journal, the Albany Law Journal of Science & Technology and Computer/Law Journal. His article, *Revising the Copyright Law for Electronic Publishing*, was published in the John Marshall Journal of Computer and Information Law in 1995. He is currently Vice-chair of the Chicago Bar Association Computer Law Committee. Some of his articles can be found on-line at http://www.Loundy.com/

Panel Summary

The rapid growth in computer network technology and on-line services continues to generate a dramatic increase in the number of networks and in the number and variety of users of these electronic communications services. Computer network security has lagged behind the implementation of new and increasingly complex computer systems. As more and more services are demanded and used by more and more people and institutions, there are more and more ways that things can go and do go wrong, which in turn give rise to consequences offending or injuring the interests or the assets of individuals and businesses. One factor which has not been given sufficient consideration, but which could become extremely important in the equation of how to go about improving the safety and security of network computing, is the obvious conclusion that a large number of lawyers are rapidly becoming computer literate and will sooner or later be ready, willing and able to assist new claimants, who are or soon will be aware of potential legal claims for the violation of their real or imagined rights, damages to their interests and real or virtual injuries, in developing new causes of action for courts to consider, all based on computer network security failures or shortcomings.

John Montjoy will give the viewpoint of a large Internet Service Provider in discussing the current legal and regulatory environment surrounding the Internet and several aspects of information system protection hardware, software and security practices and procedures.

Drawing on some of the material collected in his recently published book, Why Things Bite Back: Technology and the Revenge of Unintended Consequences, Ed Tenner will develop one or two historical technological analogies, such as the developments in medicine or the transportation industry, which gave rise to a great deal of tort litigation, as examples of how the rapid development of new technologies and the wide-spread use of them has ineluctably led to a dramatic increases in tort liability for injured claimants. Sound information system security standards and procedures can be seen to incorporate some of the same lessons which can be learned from the study of new technology adoption and litigation explosion spirals of other evolving technologies. Based on those histories, the constant upgrading of system monitoring and attention to detail will be required to take full advantage of new security technologies, while helping to reduce the number of unfortunate accidents or risks of catastrophe. Dr. Tenner will attempt to apply the lessons learned from his historical study to the problems arising from the intensification of computer networking and some predictable failures and injuries arising from the lack of compliance with adequate computer network security administrator precautions or user vigilance.

David Loundy will summarize the cases that have attempted to impose legal liability on service providers and sysadmins and then generalize from those cases about what we might expect to see as the contract and tort bars and their respective good and bad faith claimants begin to see or at least to smell the virtual blood. David will take a step into the future and discuss some of the legal problems that security systems based on encryption schemes and various systems management policies may create in the form of privacy violations for negligent disclosures, or breach of contract allegations by third parties for lost information when current or past employees can't or won't decrypt keyed information. Potential liability issues involving denial of service due to security precautions will be considered.

Fred Chris Smith will moderate the panel discussion. Drawing on his background as a litigator and as a criminal prosecutor of financial fraud and civil RICO enforcement actions involving complex criminal schemes, he will suggest that our telecommunication miracles will be just as valuable for criminal enterprises as they have proven to be for legitimate businesses. Given the almost perfect vacuum of law enforcement capabilities currently available to deal with this growing criminal problem, there is apt to be even greater pressure placed upon the traditional alternatives to criminal enforcement of financial fraud and other white collar crimes, through increased regulation and civil litigation in one form or another. In this chaotic transition from the relatively secure MIS based corporate information to open systems and global networks, legal standards are being established by negotiation, custom or by jury verdict, rather than by legislation and enforced by regulation or police action. In such a world, it is most likely that the major deterrents to attacks on network security systems will not come from public law enforcement agencies, but will be privately orchestrated and pursued, in part through an increased number of civil law suits. In the absence of a clear set of legal standards of right and wrong and lacking any reasonably certain punishment meted out by the criminal justice system, it may prove difficult for system administrators and attorneys alike to draw clear lines between unjustified civil suits based on phantom risks, and the kinds of negligent failures to comply with generally recognized standards for adequate security precautions, which should give rise to legal liability and claims for damages.

Time will be made available between presentations and at the conclusion of the discussion among the panel members for written and oral questions from the audience.

V-Chip: Policies and Technology

Moderator

Hilary H. Hosmer
Data Security, Inc.
58 Wilson Road
Bedford, MA 01730

(617) 275-8231 (fax and voice)
Hosmer@dockmaster.ncsc.mil (email)

Panelists

David Moulton
Congressman Markey's office, U.S. House of Representatives
Author of the V-Chip legislation

Susan Goering, Esq.
American Civil Liberties Union (ALCU)
First Amendment specialist

Dr. Michael Brody, MD
Child-development expert
Baltimore MD

Whitfield Diffie
Sun Microsystems
Information Security expert

Invited Speaker
Broadcaster
Entertainment Industry expert

Abstract

The U.S.A. Telecom '96 bill enables parents who use the V-Chip (V stands for Violence) to control more precisely what kinds of programming their children watch. The V-Chip is a hardware device which will be inserted into new televisions to read labels attached to programs and pass only permitted programs. The entertainment industry has agreed to come up with a labelling scheme like the current G, PG, PG-13, R, X scheme used for movies.

The INFOSEC community, because of its work with label integrity, access control, encryption, and security policies, is well-positioned to advise on V-Chip technical issues and policies. This panel provides an opportunity for dialogue with major players of diverse perspectives in the current debate.

The Telecom '96 bill is highly controversial, and portions have already been successfully challenged by the American Civil Liberties Union. Defenders are expected to take the issues to the Supreme Court.

The goal of this panel is to work toward a consensus on some of the major issues. Each person on the panel will have 5-7 minutes to make a statement, raising issues and concerns. Then the chair will raise the most interesting issues individually for discussion by the panelists and members of the audience.

Example V-Chip Issues

The sponsors of the V-Chip legislation are concerned about too much violence in our entertainment media, and the impact that this is having on our youth. Parents who don't know what is in a program can't turn off offensive programs before their children see them.

Can the V-Chip be used to turn off commercials?

Parents in favor of the V-Chip dislike the expense of buying a new television to get the capability. They worry whether their children will reprogram the device and undo parental preferences. Will parents be restricted to only the programs they permit their children to see?

Parents opposed to the V-Chip doubt that labels will improve the "vast wasteland." They have already taught their children to turn off "garbage" on the tube.

Multimedia capabilities are producing an overlap between TV and computers. Will the V-Chip be extended to handle offensive traffic on the Internet? If not, will entertainment received via computer provide a way for children to bypass TV V-Chip controls?

Foreigners are concerned that this U.S. law may have an impact far beyond USA borders. Can foreigners be prosecuted if they do not label their programming according to USA standards and it gets to the USA? If not, will U.S. entertainment providers move offshore?

The entertainment industry is concerned about censorship and the impracticality of rating thousands of hours of daily TV fare. Labeling is a major issue.

> Whose standards are to be used in doing the labeling? New York or Kansas?
>
> Should sex and violence be rated orthogonally?
>
> Should historical violence and fictional violence be treated the same way?

If some portions of a program are violent, should the entire program be labeled at the level of the most violent scene in the movie? Or should different parts be labeled differently, so that only offending portions are blocked by the V-Chip?

What will guarantee the integrity of the labels?

PROTECTING MEDICAL RECORDS AND HEALTH INFORMATION

Panel Chair
Joan D. Winston, Trusted Information Systems, Inc.
Panelists
Gail Belles, VA Medical Information Security Service
Bill Braithwaite, U.S. Department of Health and Human Services
Paula J. Bruening, Information Policy Consultant
Patricia Taylor, U.S. General Accounting Office

Patient care is becoming increasingly computer-intensive. Electronic records and transactions are central to health-care administration, payment, and cost containment. The prospect of lifelong electronic patient records -- whether stored centrally, in geographically distributed but logically linked databases, or in portable tokens -- is just below the horizon. As a result, public and government awareness and concerns regarding privacy protections for these records is also increasing. Congress is attempting to establish a new privacy framework for medical and other personal information in electronic, networked environments. At the same time, despite severe resource constraints, Federal agencies must meet the challenges of safeguarding health-related information for tens of millions of Americans.

This panel will examine the technical, policy, and legal issues involved in establishing and implementing appropriate protections for patient medical records and other types of health information.

Audience participation and discussion will be encouraged! Topics that we will explore include:

- Information security principles and practices in the patient-care environment;
- New medical-records issues presented by networking;
- Controversies over secondary and unanticipated uses of health information held by the public and private sectors;
- Impact of U.S. and OECD information privacy laws and policies, including new legislation;
- Federal agency approaches to medical record and health information protection;
- Evaluation of privacy and security implementations;
- Impact of cryptography policies on medical record and health information protection.

PROTECTING MEDICAL RECORDS AND HEALTH INFORMATION

Contact Information for Panelists

Gail Belles
Acting Director
Medical Information Security Service
Veterans Administration Medical Center,
Bldg. 203B
Martinsburg, W. VA 25420
Voice: 304/263-0811 X 4077
Fax: 304/264-4497
<gab@intrepid.net>

Bill Braithwaite
Senior Advisor on Health Information Policy
Office of the Assistant Secretary for Planning and Evaluation
U.S. Department of Health and Human Services
Room 440-D Humphrey Building
200 Independence Avenue, SW
Washington, DC 20201
Voice: (202)260-0546
Fax: (202)690-5882
E-mail: BBraithw@osaspe.dhhs.gov

Paula J. Bruening
Information Policy Consultant
3525 Davenport Street, NW #505
Washington, DC 20008
Voice: 202/966-1805
<pjb@crosslink.net>

Patricia Taylor
Director, IRM/HEHS
Accounting and Information Management Division
U.S. General Accounting Office
Washington, DC 20548
Voice: 202/512-5539 or 512-6408
Fax: 202/512-6451

Joan D. Winston*
Principal Policy Analyst
Trusted Information Systems, Inc.
8000 Westpark Drive, Suite 600
McLean, VA 22102
voice: (301) 854-6889
fax: (301) 854-5363
<jwinston@tis.com>

* Address after August 1, 1996. Prior to August 1, contact information is: Trusted Information Systems, Inc., 1420 Spring Hill Road, Suite 600, McLean, VA 22102; voice: (703) 917-6630; fax: (703) 821-8426.

<u>Panelist Outline: Protecting Medical Records and Health Information</u>

Gail Belles, Acting Director
Medical Information Security Service
Veterans Health Administration

- Brief background of VHA's health care system and automated hospital information systems
- Changing technologies—shift from mainframe to distributed computing environment
- Goals of information security—laws and standards that impact these goals
- Standard security controls in place
- Specific security controls in medical record applications
- Kernel System Management
- Network protection issues
- Current projects/information security issues

Crimes in Cyberspace: Case Studies

Moderator
William S. Galkin, Esq.
Law Office of William S. Galkin
10451 Mill Run Circle, Suite 400
Owings Mills, Maryland 21117 U.S.A.
tel: 410-356-8853
fax: 410-356-8804
email: wgalkin@earthlink.net

Panelists
Arnold M. Weiner, Esq.
Weiner, Astrachan, Gunst, Hillman & Allen
Baltimore, Maryland

Kenneth C. Bass, III
Venable, Baetjer, Howard & Civeletti
Washington, D.C.

Abstract

The vastness of Cyberspace offers numerous opportunities for criminal activities. Crimes may include fraud, copyright infringement, cyberstalking, intrusions into computer systems, privacy violations, industrial espionage. Cyberspace also offers new opportunities to facilitate many of the more "traditional" crimes. The panel will present, discuss and analyze the legal issues involving several actual criminal incidents the have occurred in Cyberspace.

CURRENT CHALLENGES IN COMPUTER SECURITY PROGRAM MANAGEMENT

Panelists:

Barbara Guttman
National Institute of Standards and Technology
Building 820, Room 426
Gaithersburg, MD 20899
e-mail: barbara.guttman@nist.gov

Lynn McNulty
McNulty and Associates
P.O. Box 6101
McLean, VA 22106
e-mail: LYNN.McNULTY@INTERNETMCI.COM

Paul M. Connelly
Chief, Security and Safety Division
White House Communications Agency
The White House
Washington, D.C. 20500

Ann F. Miller
Fleet and Industrial Supply Center
Code 12/80.1
1968 Gilbert Street
Norfolk, VA 23511-3318
e-mail: ANN_MILLER@WP-EMH1.NOR.FISC.NAVY.MIL

Mark Wilson (Panel Chair)
National Institute of Standards and Technology
Building 820, Room 426
Gaithersburg, MD 20899
e-mail: mark.wilson@nist.gov

PANEL SUMMARY

Managing a computer security program has been getting more difficult in light of budget constraints, reorganizing and downsizing, and the continuing decentralization of ever-increasingly complex computing and communications environments. This panel will discuss the changes in OMB Circular A-130 and the document's impact on computer security programs, the marketing of a computer security program, how to build a successful program, how to keep a program stable during unstable times - during a reorganization, and the efffective use of collateral-duty personnel to support and augment the computer security staff.

Barbara Guttman, NIST - A new version of OMB Circular A-130 was signed on February 8, 1996. The Circular provides uniform government-wide information resources management policies, including computer security policies. The main thrust of the new version of the Circular is to drive security responsibilities down to the users and managers of computer systems and information. To address computer security in today's environments, users and managers need a framework which can handle a myriad of technological possibilities. The Circular suggests a structure with two categories: general support systems and major applications. Another important change in the structure is that the new A-130 does not distinguish between "sensitive" and "non-sensitive" systems. These and other changes will be discussed.

Lynn McNulty, McNulty and Associates - Knowing the computer security requirements is only the beginning. A newly-appointed computer security officer, or a veteran in a newly-established computer security program must to be able to convince often-sceptical agency executives, managers, and users that computer security is important, why it is important, and why computer security needs to be integrated into the agency's business and decision-making process. Useful strategies for working with these audiences, getting others to accept the responsibility for "doing" computer security, as well as how to better your chances of winning budget and people battles will be discussed.

Paul M. Connelly, White House Communications Agency - This presentation contains first-hand examples of how a successful computer security program was built using these strategies. Topics that will be discussed include how a security program was built from scratch to protect some of our nation's most sensitive and critical information systems in a highly operations-driven environment, and obtaining management buy-in (e.g., identifying key allies, involving management in setting program goals and priorities, and obtaining management committment for specific objectives). The speaker will also address what worked, what did not work, obstacles faced, and will offer a recipe for success.

Ann F. Miller, Fleet & Industrial Supply Center, Norfolk (Department of the Navy) - Once a computer security program matures, it still faces a number of pitfalls. One challenge that program managers face today is keeping a successful security program intact during a reorganization, or a series of reorganizations. Topics including policy and procedure enforcement, changing management structures and reporting paths, establishing security agreements between new or changing organizations, inspections and compliance checks, preparing for inspector general (IG) visits, and keeping up with the changes to the network of collateral-duty security personnel will be discussed.

Mark Wilson, National Institute of Standards and Technology - One tool in the computer security officer's toolkit to meet today's funding challenge is the effective use of collateral-duty security personnel. Some agencies have found that a network of collateral-duty personnel, appointed for each network and system in an agency, makes implementation and maintenance of policy, procedures, and practices more manageable, negates the possible impact of distances between some agency offices, and provides easier individual identification and auditability for the computer security officer. Utilization of this approach can help spread the workload more evenly among system users and system administrators. This can also increase the agency-wide awareness of information systems security responsibilities, while utilizing the existing management structure.

Panel: Achieving Vulnerability Data Sharing

Researchers in communities including intrusion detection, security, incident handling, and software engineering have long expressed an interest in having access to a repository of vulnerability data that could be used in their experiments and analyses. These communities have different requirements for such a repository and would derive different benefits from it. These differences have often been cited as obstacles to the creation or sharing of such a repository.

Issues that have been defined in building a repository include:

- determining a vulnerability classification scheme,
- defining useful levels of abstraction for vulnerability definition for research, incident handling or intrusion detection,
- developing the data structures and applications to support the classification scheme,
- developing a sanitization method that protects incident victims,
- ensuring the integrity and authenticity of the repository data,
- regulating access to the data to only those with legitimate need, proprietary constraints, and other external controls (and defining what "legitimate need" might be).

Other administrative issues to be addressed include the collection and dissemination qualifications among the users, overall management of the repository, and resource requirements. Broader issues would include unanswered legal questions regarding participation and information dissemination, and participant trust limitations.

This panel will discuss some of these issues based on lessons learned from ongoing efforts to promote more data sharing within and among these communities.

The panel participants include:

Lisa J. Carnahan, Panel-Chair
National Institute of Standards and Technology

Matt Bishop
University of California, Davis

James Ellis
CERT Coordination Center

Ivan Krsul
COAST Laboratory, Purdue University

INCIDENT HANDLING POLICY, PROCEDURES, AND TOOLS

As more organizations connect to the Internet and share information globally, the need for a rapid incident handling capability increases. The number of Internet related incidents that have occurred in the past year require organizations to take seriously their incident handling capability. The Office of Management and Budget has reinforced this need by requiring in the newly revised OMB Circular A-130, that federal agencies be able to respond in a manner that both protects its own information and helps to protect the information of others who might be affected by the incident. This new requirement comes at a time when federal agencies are being faced with reduced budgets and staff. Many organizations have already developed incident handling teams or incident handling procedures. This panel will discuss the incident handling policy and procedures that have been implemented within their organizations. In addition, a new methodology that system administrators can use for characterizing network security tools will be discussed.

Chair:
Marianne Swanson, National Institute of Standards and Technology

Panelists:
Kelly Cooper, BBN Planet
BBN Planet is an Internet Service Provider. As a service provider they notify their customers of major security events and problems and field calls from customers asking for information on and help with incidents. When an incident is reported, BBN's goals are (a) to perform identification of the problem (to confirm whether the situation is a security incident and determine the seriousness of the problem), (b) to do damage control (i.e. making router filters more restrictive or taking the customer off the net until they have repaired their breach), and © to provide encouragement to the customer in contacting other sites involved. They also provide basic information on policies/procedures and direct customers to CERT's patches, tools and information. The practices and procedures that BBN uses in their incident handling efforts will be presented.

Thomas Longstaff, Computer Emergency Response Team/Coordination Center (CERT/CC)
This presentation will report on a new methodology for characterizing the capabilities of network security tools. This method determines what threats or risks are addressed by the collections of tools in a network environment. In particular, for each tool identified, it is possible to determine what the tools do and what threats or risks are addressed. From this assessment, a network administrator will be able to determine which risks are managed appropriately and use the result as a guide for acquiring new network security tools. In addition, it will be possible to use the method to determine if existing tools cover newly discovered vulnerabilities or if a newly developed tool will cover additional threats. Unlike an evaluation of a security tool, this method does not address the "goodness" of the security tool, but only its designed capabilities. As a practical example, we will provide the results of applying the methodology to a representative set of existing tools to identify what threats they cover in a network environment.

Peter Richards, Westinghouse Savannah River Company
The Department of Energy's CIAC response team is used to augment Savannah River's incident handling capability. Many organizations are employing outside assistance to handle incidents if they become too large in scope or too difficult to handle in house. This type of incident handling support is being implemented more frequently. The policy and procedures that are used in this company will be discussed.

Ken van Wyk, Science Applications International Corporation (SAIC)
SAIC's Security Emergency Response Center (SERC) provides fee-for-service assistance to commercial and government organizations in need of on-call and on-site security incident response support. The procedures that SERC uses to handle their client's incidents will be reviewed.

Marianne Swanson, National Institute of Standards and Technology
National policy now requires agencies to develop an incident handling capability. NIST has been tasked to facilitate incident handling for the federal agencies by providing standards, guidance, and mechanisms for sharing information. The status of NIST's progress in this area will be presented.

Panel

Interdisciplinary Perspectives on Information Security: Mandatory Reporting

Panel organized by the National Computer Security Association
M. E. Kabay, Ph.D., Director of Education

Background:

The National Computer Security Association (NCSA) is dedicated to enhancing communications among providers and users of information security technology and knowledge. Our special interdisciplinary symposium this year is mandatory reporting.

Problem:

The information security profession lacks a factual basis for estimating the extent, methods and costs of computer crimes and accidents. The problems of ascertainment are that

(1) there is evidence that most computer crimes and accidents are not detected at all;

(2) most of the detected crimes and accidents are never reported to anyone.

This gross lack of data interferes with professional efforts to alert upper management of the importance of improving information security and prevents rational allocation of scarce corporate and national resources.

Proposal:

A mandatory reporting system with full guarantees of confidentiality and anonymized data records would track occurrences of crimes and accidents and provide statistical reports and case studies. Such a data-gathering and -reporting agency would complement the purely technical records of the CERT-CC and other agencies that collect data on computer crime and accident but rarely publish detailed case studies. It would provide a growing basis for sound financial decisions on allocation of resources to different protective measures.

Purpose of Symposium:

Panelists will discuss the experiences of their own disciplines with mandatory reporting with an eye to our avoiding known pitfalls and benefiting from their years of experience. Each of four panelists will have up to 20 minutes to present a review of mandatory reporting in their own field. The panel will then be open for questions from the audience.

Panelists:

U.S. Agency representatives familiar with the history of mandatory reporting of problems in their fields will explain their agencies' experiences in this difficult endeavor:

- Bruce Butterworth, Director, Office of Civil Aviation Security, Federal Aviation Administration;

- Barbara Smith Jacobs, Chief, Office of Disclosure Policy, Division of Corporation Finance, Securities and Exchange Commission;

- Bob Whitmore, Chief, Record Keeping Requirements Division, Occupational Health and Safety Administration;

- Dr. Scott Wetterhall, Acting Director, Division of Surveillance & Epidemiology, Centers for Disease Control and Prevention.

The challenge for our speakers is to identify the historical sequence of development of mandatory reporting and to focus on lessons for the information systems security field. We need information about

- how to build support for mandatory reporting among the affected organizations;

- what lessons have been learned about minimizing inconvenience to the organizations affected by mandatory reporting;

- privacy / confidentiality issues and how they have been handled;

- helpful statistical approaches to reporting results to the affected organizations and to the public;

- avoiding key errors in getting the whole process of mandatory reporting on a sound footing.

The NCSA invites everyone interested in the possibilities of mandatory reporting of information systems security breaches to participate in the symposium and to join in the discussion and debate that will follow the formal presentations by our speakers.

INTERNATIONAL PERSPECTIVES ON CRYPTOGRAPHY POLICY

Panel Chair
Dorothy E. Denning
Georgetown University, Computer Science Department
225 Reiss Science Building, Washington, DC 20057-1232

Panelists
Peter Ford
Attorney General's Department, West Block Offices
Queen Victoria Terrace, Parkes ACT 2600, DX 5678 Canberra, Australia

David Herson
Commission of the European Communities, Directorate-General XIII
Rue de la Loi 200, B-1049 Brussels, Belgium

Nigel Hickson
Department of Trade and Industry, Policy for IT Security
151 Buckingham Palace Road, London SW1W 9SS, U.K.

Panel Summary

Panelists from outside the United States will discuss their views on cryptography policy and national and international proposals and initiatives. Efforts within the Organization for Economic Cooperation Development (OECD) to write cryptography policy guidelines will be reviewed. The panelists will describe initiatives to establish a cryptography infrastructure within their countries and internationally to support the security needs of the global infobahn. They will discuss the role of trusted third parties or key escrow in encryption policy and infrastructure services, and issues that need to be resolved.

INTERNATIONAL PERSPECTIVES ON CRYPTOGRAPHY POLICY: A UK PERSPECTIVE

Nigel Hickson
Department of Trade and Industry, Policy for IT Security
151 Buckingham Palace Road, London SW1W 9SS, U.K.

The United Kingdom authorities have, in common with other administrations, been working for some time to develop policies, concerning the use of encryption, which balance the ever increasing industry requirements (for strong security) with national law enforcement needs. Our efforts have been intensified by the requirements of UK businesses to play a full role in the emerging information society, which include being able to take part in global electronic commerce.

After two years of discussion Government Ministers have now committed themselves to an encryption policy which has, as its centre-piece, the licensing of Trusted Third Parties (TTPs) who will enable their clients (individuals or business) to have access to a number of different cryptographic services. The formal announcement, to Parliament, by the Science and Technology Minister, Ian Taylor, is given below. The most important service a TTP can offer, will, we believe, be integrity. An infrastructure whereby the public encryption keys of business can be verified and authenticated is urgently required to enable business to engage in commerce with companies with a degree of trust. In this context we hope that digital signatures will be one of the services a TTP may be able to offer. We also recognise that confidentiality, of both stored information and that which is transmitted, is becoming increasingly important for business. We therefore envisage that the TTPs will, in conjunction with IT suppliers, offer a key escrow service to allow their clients to converse securely with all other TTP clients on the "network". For law enforcement the TTPs - which will be licensed by Government - will be required to supply their client's private encryption keys to Government under due legal process. Apart from the latter there will be no new controls on the right of any company or individual to use encryption technologies in the UK.

There is, however, little point in establishing TTP networks solely in the UK. The needs of business (nor the communication networks to support them) are not restrained by national borders, and therefore to facilitate global commerce the "network" of TTPs (briefly referred to above) will need to be established. This will take time, will involve much policy discussion in the EU, OECD and other bodies, and will only happen if business wants it to. The UK have but taken a small first step.

ENCRYPTION POLICY ANNOUNCEMENT: PARLIAMENTARY ANSWER ON 10 JUNE 96

Ian Taylor MP, Minister for Science and Technology

Following the discussion between Departments to which I referred in my replies to the hon Member for Brigg and Cleethorpes of 6 and 25 March, I am today publishing a paper outlining the Government's policy on the provision of encryption services on public networks. Copies of the paper are available in the library of both Houses.

The Government aims to facilitate the development of electronic commerce on the emerging global information infrastructure. This is of significant importance in maintaining the UK's competitiveness and is a component of the department's information society initiative. There is a growing demand for encryption services to safeguard the integrity and confidentiality of electronic information transmitted on public telecommunications networks. The Government therefore proposes to make arrangements for licensing Trusted Third Parties (TTPs) who would provide such services. These TTPs would offer digital signature, data integrity and retrieval, key management and other services for which there is a commercial demand. The Licensing policy will aim to protect consumers as well as to preserve the ability of the intelligence and law enforcement agencies to fight serious crime and terrorism by establishing procedures for disclosure to them of the encryption keys, under safeguards similar to those which already exist for warranted interception under the Interception of Communications Act.

Officials from within within my department have held preliminary discussions with industry groups on the concepts set out in the paper. The Government intends to bring forward proposals for legislation following consultation by the Department of Trade and Industry on detailed policy proposals.

Panel

Security Protocols/Protocol Security

Moderator

Doug Maughan

INFOSEC Research Division

National Security Agency

(301) 688 - 0847

Panelists

Representatives from Industry and Consortia working in the secure standards and protocols arena

Abstract:

The burgeoning use of the Internet for commercial traffic, as well as increased personal use of computers in the home and for business has meant a continuing increase in the need for security. Because of travelling users, client/server and object-oriented architectures, and heterogeneous business systems, standards and protocols for the net are critical to continued growth. Since security is also a foundation for these sorts of Internet use, security in the protocols, and security considerations in the standards and standard processes are also critical. This panel will discuss various protocols and standards related to security, assess their importance and usefulness, and the potential for narrow or widespread adoption of these standards.

SURVIVING THE YEAR 2000 TIME BOMB

Moderator
Dr. Grace L. Hammonds

AGCS, Inc.
91 Montvale Avenue
Stoneham, MA 02180-3616

TEL 617-279-2864
FAX 617-279-2865

Hammonds@Dockmaster.ncsc.mil

Panelists
James W. White
National Director of the Millenium Solutions Center
OAO Corporation

Andrew Hodyke
United States Air Force ESC/AXS
Software Design Center

ABSTRACT

As the century, and the millennium, is coming to an end, the world's computer systems may quite literally have time bombs getting ready to go off. At the core of the problem is the all too common practice of storing and processing dates with a two digit "year in century". Since date calculations are pervasive in information systems, we can expect widespread ramifications -- and the information security arena is not spared. Compounding the problem, many date algorithms will not properly recognize year 2000 as a leap year. This panel will identify the complexity and magnitude of the Year 2000 Problem, why so many people will likely be affected, and some practical near and long-term solutions.

SURVIVING THE YEAR 2000 TIME BOMB

What's the Problem?

The year 2000 will mark the first century change since the computer revolution began. Because of this, it has been and is still common practice for computer-based representations of the year to use a 2-digit encoding, and assume the first two digits are '19' (as in MM/DD/YY). As a result, the year 2000 and 1900 become equivalent. The impact is not only in on-screen or stored representations. Any date calculations that do not account for the overflow (in this cases to two places--from 1999 to 2000), will become miscalculations. Such date errors affect sequences, time spans, durations, schedules, and a host of other related information.

At the root of this problem is that time is a continuum -- constantly moving forward -- so in theory, later dates should always be represented by larger numbers. In practice, computer representations are necessary limited – by allocated field widths internally and by fixed screen and printer formats when viewed by users. As we approach the year 2000, two-digit date fields will wrap, and as a result look as if we've moved back in time.

By the way, this problem can be expected to crop up earlier than January 1, 2000, because any systems that calculate dates into the future could be vulnerable (e.g., in Massachusetts, car registrations are valid for four years).

A second problem looming with the century (and millennium) change involves leap years. Many of us have always believed that a year is exactly 365.25 days, so we simply added an extra day to February every four years. In fact, a year has been calculated at slightly below that [FAQ96], so every fourth year is a leap year, EXCEPT every hundred years, EXCEPT every 400 years. So,

1896 is a leap year (divisible by 4)
1900 is not a leap year (divisible by 100)
2000 is a leap year (divisible by 400)

Whether to add the extra day to the year has been an easy problem for the last 100 years, but that will soon change. In many cases, the result will be incorrect date representations after February 28, 2000.

Why did this happen? This practice of the two-digit year began in part due to limited space for data storage and display screens. And of course, when there were 30-40 years left in the century, date calculations did not seem to be in any danger. Basically, systems and software have exceeded their expected software life cycle. How could programmers know their software would last this long!

The impact to systems both in the US and around the world is already being assessed. Recently [US96], the US House of Representatives held a hearing on this problem, inviting government and business representatives to present the issues. Financial institutions, medical centers and insurance companies, even transportation systems (because of schedules) could be staggering. If these calculations are used in process control or for real-time systems, say at a nuclear power plant, the ramifications can be considerably more serious.

What's the Security Problem?

The affect of this faulty date logic on system and data integrity can be expected to be widespread and costly. If this problem is not addressed, we can also expect to lose systems, as they come crashing to a halt or simply don't start. [IBM96] [FAQ96] [1]

At the lowest level, system software maintains internal clocks that have already been known to fail when the field containing the time overflows [NEU95].

Timestamps are used widely, particularly in audit trails. Even backups and archives are at risk [DA96], since incorrect retention records could result backups being deleted prematurely. Some mainframe libraries use the two-digit year as part of their labels, and base their tape retention logic on label calculations.[2]

Timestamps also appear prominently in cryptographic data, in key validity periods and expiration times. Some hashing and random-number generators use parts of the system date as a parameter.[DATA96]

Logins could be affected if password expiration logic is faulty.

Application software could mysteriously stop if license period calculations fail.

At a management level, if in trying to address this problem one brings in outside consultants to analyze information systems, in the process, confidentiality of the data could be sacrificed.

National security could even become an issue. Within DoD [US96], there are thousands of computers, some of which are custom-designed and control weapons (including DoD-unique computer chips that are no longer being manufactured). Custom software is widely used (the older systems are sometimes referred to as "legacy"). These applications were designed with multiple languages, and for some, compilers and even programmers are scarce or no longer available. The problems range across the board from office systems (finance, accounting), to logistics, to command and control, to weapons systems. On top of this, the problem must be addressed in the face of declining defense budgets.

---

[1] IBM has acknowledged that it will have a problem with certain versions of VM and initial program loads (IPL) in 2000. It is reported that IBM will not fix the software in some of its older systems. Users of the old System 370 machines will just have to upgrade. [IBM96] [FAQ96]

[2] The identifier situation is not unusual. Government contracts often appear with two-digit dates as part of the control number, which could become a problem if the year field is used, say for sorting.

Why You Should Care?

If this is not already enough to convince you to care, consider the following.

Tens of millions of PC users likely have a system with the century rollover problem. Many PC's have a problem with the BIOS logic. The CMOS real time clock will fail to properly maintain the year after December 31, 1996. Variations on the problem have also been recorded in Windows 3.1 and Windows 95.[DA96]

You may think that by 2000, you will probably replace your own computers, and your OS software (possibly several times), so the hardware and system software vendors will probably take care of the problem. But your hardware is only the tip of the iceberg.

Your application software may also have faulty date logic, incorrectly using two digits for year calculations. If it was specially designed for your business, you may want to continue to use it. Also commercial (packaged) software is not immune. The problem has also been reported in date functions of dBASE III+.

Even if you are willing to dig into your software yourself, bear in mind that with commercial software, usually no source is available for your review (let alone to repair). In some cases, vendors may no longer exist.

To make things even more interesting, Jan. 1, 2000, falls on a Saturday.

What You Can Do?

Replacement of all software and hardware over the next few years is not necessarily the only option, although it may seem the most obvious one. Some of the solutions lie in:

Computer hardware and software manufacturer initiatives to upgrade their systems to be "year-2000 compliant"

Standards organization efforts to move to four-digit years

Resources on the internet and elsewhere to stay in touch with Year 2000 developments

Tools and services to help assess individual problems and make appropriate changes

Approaches for generally including year-2000 upgrades in the normal maintenance routine for systems

Dealing with this is one project that cannot slip -- January 1, 2000, is a hard and fast deadline.

References

[IBM96] Link from IBM WWW, http://www.ibm.com.

[DA96] Datamation Special Report on the Year 2000 Problem. January 1, 1996.

[FAQ96] Frequently Asked Questions, link from WWW, http://www.year2000.com.

[NEU95] Peter Neumann, Computer Related Risks, Addison Wesley, Reading, MA, 1995.

[US96] Statement by the Honorable Emmett Paige, Jr., Asst. Secretary of Defense (C3I) before the Committee on Government Reform and Oversight, Subcommittee on Government Management, Information and Technology, US House of Representatives, April 16, 1996.

Panel

Database Systems Today: Safe Information at My Fingertips?

Chair

John R. Campbell, NSA

Panelists

Tim Ehrsam, Sr. Product Manager, Security Products, Oracle Corp.

Dick O'Brien, Secure Computing Corporation

Thomas Parenty, Sybase Corporation

LTC Ken Poindexter, DISA

Satpal S. Sahni, 3S Group Incorporated

Informix, TBD

Introduction

Overnight, it seems, we are able, with our net browser or database client, to access data from anyplace in the world. We have both internet and intrnet access. The architectures now available include client/server and distributed and may included firewalls, web servers and wireless connections. Unknown to us, we can be gathering information from many servers, owned and controlled (or not controlled) by many different entities.

The database server is also changing. The relational engine is either gaining new capabilities or is having other engines added to it to handle text, multimedia and temporal data. Object-oriented database systems are becoming more common and are better able to handle both object and relational data, and other products are combining relational and object capabilities. More and more systems are multivendor, bringing with them the possibilities of inconsistencies, including security inconsistencies, that multi-vendor systems bring.

How do I know that my web browser is properly interfacing with database servers that are located thousands of miles from me? How do the database servers know, especially if I use the Internet, that I should be looking or entering data into the server?

Database vendors are looking at some of the security issues, including strong identification and authentication, data confidentiality, data integrity, single signons, mutual authentication, SQL pass-through/control on firewalls, web/DBMS server data passing security and wireless connections. Application builders are constructing very useful systems consisting, for example, of web browsers, servers and multilevel database systems. They also are looking at the wider use of encryption and are eagerly looking at internet electronic commerce.

The good thing about many of these activities is that we are getting lots of functionality and ease of use. A problem is that we may be endangering the security of our data. This panel will address the following questions:

1. What new capabilities are the database vendors offering?

2. How do these capabilities affect the security, including data integrity, of the data?

3. What are the vendors doing to solve the security problems?

4. What are others doing to supplement the user's needs for security?

5. What applications are being built to satisfy the new user requirements?

6. What security issues still need to be addressed?

We are fortunate today to have a distinguished panel to address these questions. Tim Ehrsham, Thomas Parenty, and ___ are very senior security technologists in database firms that dominate their industry. Dick O'Brien is an internationally known expert in high assurance database systems. Satpal Sahni is an expert in high assurance Identification and Authentication and has used this expertise to implement an operational system. Finally, LTC Ken Poindexter is leading an effort to build an important, innovative, new information system that satisfies the needs of ease-of-use, separation of information and data security. Please hold questions until the end of the session.

Webware: Nightmare or Dream Come True?

Chairman
Peter G. Neumann, Computer Science Lab, SRI International

Panelists
Steve Bellovin, AT&T Research
Ed Felten, Department of Computer Science, Princeton University
Paul Karger, IBM Thomas J. Watson Research Center
Jim Roskind, Netscape Communications Corporation

Panel Overview

Peter G. Neumann
Computer Science Lab, SRI International
Menlo Park, California

This session considers the risks involved in the open-ended runtime security problem introduced by world-wide web browsers and programming languages such as Java and JavaScript, as well as other languages with similar problems -- such as ActiveX, Microsoft Word macros, and PostScript. [1] The ability to execute arbitrary code of unknown trustworthiness from unknown sites (perhaps without your awareness) presents many fascinating security challenges. The session will explore various approaches to avoiding or living with those risks. This problem has the potential for greatly advancing computer use if it is handled intelligently, and greatly impairing security if it is not.

In this context, runtime security depends in many subtle ways on operating systems, networking, applications, programming languages, bytecode problems, browser design and implementation, the interfaces between languages and browsers, cryptographic embeddings, interoperability constraints, backward-compatibility requirements, user interfaces, security policies, tradeoffs between static and dynamic checking, and many other factors. Possible techniques for increasing security include cryptographic signing, unconstrained sandboxing, confined sandboxing, trusted computing bases, firewalls, starkly restricting programming language capability, draconian interpretive execution, and so on. Other techniques may be desirable for monitoring usage and periodically removing undesirable residues. None of these techniques is adequate in isolation, and the overall problem requires total system approaches to provide any assurance whatever of being able to avoid or dramatically reduce the risks.

We must urgently anticipate the future, because the problems are not getting any easier and the risks are expanding as increased application demands are placed on our systems and networks. The following position statements represent a very interesting cross-section of the spectrum of approaches, and will undoubtedly lead to lively and challenging discussion.

---

[1] Many terms used in these viewpoints are trademarks or registered trademarks of their owners. Where a trademark is known, the term has been capitalized.

Viewpoint

Java -- Threat or Menace?

Steve Bellovin, AT&T Research
Murray Hill, New Jersey

Java and its kin (ActiveX, MS Word macros, even Postscript) are a new and potent threat to computer security. To a great extent, these systems are not secure, and cannot (on today's hosts) be made secure without crippling their essential functionality.

The problem is that we (They) want to be able to execute programs with many privileges and abilities, but still prevent these programs from doing the Wrong Thing. Unfortunately, the containment technology is often not up to par to deal with the threat.

While this note focuses on Java, much of it would apply to many – and arguably most -- similar systems.

First of all, there is a Java security model -- a paper list of what Java applets should and should not be able to do. The standard list of permitted operations includes network I/O to the source host, and file I/O to a restricted list of files. At this point, we are already in trouble -- empirically, it's hard to parse file names correctly (there are many examples of such failures going back at least 15 years), users can be tricked into changing the list (and there's no system-settable default), and there is no higher-level protection mechanism (such as the OS kernel) that can be brought to bear.

Network I/O is another problem. For one thing, it implies DNS query ability, because the permitted destinations are defined by name, not address; this in and of itself can be used to leak information. (We won't even discuss the fact that that check as implemented has had security problems.) Worse yet, the network abilities can be used to attack some firewalls from the inside. Firewalls are designed on the assumption that the bad guys are on the outside; they do not, as a rule, assume that insiders are trying malicious things. This conflict of models, between the firewall and Java, causes trouble -- a Java applet, behaving within the strict limits of the nominal security model, can poke a hole through a firewall which is itself behaving properly. The fundamental cause of this problem is the attempt to compose security policies -- the result is itself not necessarily secure.

On a deeper level, the Java security policy is implemented in terms of the Java language definition. This definition has not, to my knowledge, been formally verified. Bugs in the definition can lead (and indeed, have lead) to holes. Furthermore, the user never sees the Java source; rather, a "byte code" is downloaded for execution. A verifier -- that is, a theorem prover -- attempts to check if the byte code has legal semantics. Here, we must first assume that the theorems are correct, and second, assume that the verifier is in fact checking correctly. Neither assumption seems warranted.

There is a vast contrast between the complex security model relied upon by Java virtual machine and the relatively simple security model used by real hardware. The latter checks a few bits per page to regulate storage access, and a few bits in the machine state to limit use of privileged instructions. Requests for privileged operations are gated through another simple mechanism, such as supervisor calls or traps. Java, by contrast, treats all bytes as equal; everything has to be working properly to prevent inappropriate use of the so-called native methods.

For some applications, little of this would matter. After all, we regularly run -- and trust -- all manner of software. The key difference is that applets are run implicitly, without user knowledge or consent. Digital signatures would help, of course, but a signed virus is still a virus. And I despair of any scheme that requires users to consent explicitly to each applet -- the human factors are daunting.

Viewpoint

Language-based Protection: Why? Why Now?

Edward W. Felten, Drew Dean, Dan S. Wallach
Dept. of Computer Science, Princeton University

Today's executable-content systems typically use language-based protection rather than hardware-level protection. This is an idea that fell out of favor twenty years ago but is now suddenly returning. Pragmatism forces implementers to choose software-based protection for several reasons. First, hardware-based solutions are not portable across platforms. Second, commodity operating systems do not provide sufficiently flexible access to protection hardware. Third, commodity microprocessors require a significant performance penalty for crossing protection boundaries; given the fine-grain sharing exhibited by current executable content, hardware-based protection would be too slow on today's architectures [1].

Given the enormous market demand for executable content, and the apparent necessity of language-based protection as an implementation method, it appears that our security will rely on language-based protection whether we like it or not. This approach was at best an interesting failure the last time it was tried. What have we learned in the meantime? Are we doomed to fail again?

Twenty years of computer science research lead us to believe that language-based protection has a better chance of success now. First, our improved understanding of programming language semantics and related proof methods allow researchers today to prove interesting theorems about real languages [2,3]. Second, better understanding of how to support abstract data types at the language level [4] allows programmers to control access more reliably. Finally, our understanding of separate compilation, dynamic linking, and software engineering in general has matured. All these factors increase our confidence in language-based protection.

We should also recognize that the implementation of hardware-based protection has gotten much more complicated. On commodity microprocessors such as the Pentium Pro, performance pressures have forced the designers of memory-protection hardware to use advanced implementation techniques [5]. Protection-checking hardware is pipelined and the results of protection checks are explicitly and implicitly cached at several places in the chip. The resulting TCB is much larger and harder to verify. Many hundreds of thousands of transistors are used to implement memory protection and caching on the Pentium Pro. Even seemingly trivial components like the processor/memory and I/O connections are extremely complex; an informal specification for the EISA bus is over 400 pages in length [6]. The assumption that software is necessarily much more complicated than hardware, if ever it was valid, is not valid any more.

The main difference between hardware and software is that today's hardware designers don't have the flexibility to patch broken systems after they're shipping, so they increasingly use validation techniques to get it right the first time. Software designers can get away with penetrate-and-patch, so that's how they operate. If software bugs were as devastating to the vendor as hardware flaws like the Pentium FDIV bug, software vendors would learn to spend the effort to verify their systems.

[1] Thomas E. Anderson, Henry M. Levy, Brian N. Bershad, and Edward D. Lazowska. "The Interaction of Architecture and Operating System Design." Fourth ACM Symposium on Architectural Support for Programming Languages and Operating Systems, 1991.

[2] Robin Milner and Mads Tofte. Commentary on Standard ML, MIT Press, 1991.

[3] Myra VanInwegen. The Machine-Assisted Proof of Programming Language Properties, Ph.D. Thesis, University of Pennsylvania, 1996.

[4] J. C. Reynolds. "Types, Abstraction, and Parametric Polymorphism." 1983 IFIP Conference.

[5] John L. Hennessy and David A. Patterson. Computer Architecture: A Quantitative Approach, second edition. Morgan Kaufman, 1996.

[6] BCPR Services, Inc. "EISA Specification, Version 3.12." 1992.

Viewpoint

Untrusted Applications Need Trusted Operating Systems

Paul A. Karger
IBM Corporation Thomas J. Watson Research Center
P.O. Box 704, Yorktown Heights, NY 10598

The security issues of languages such as Java, Javascript, ActiveX, PostScript, etc. are not new. The desire to run untrusted applications goes back to the earliest days of computer security. The original Anderson panel report [1] describes an extremely limited subset of GECOS Time-Sharing FORTRAN that was intended to encapsulate untrusted applications and allow them to run safely on sensitive DoD computer systems. Even though the subset eliminated most of the useful features of FORTRAN, Anderson was still able to easily break out of the subset language, exploit a vulnerability in the underlying operating system, and gain fully privileged status on the GE-635 computer. This was the equivalent of gaining root access on a modern UNIX system. Except that FORTRAN was compiled, rather than interpreted, this scenario bears a remarkable resemblance to downloaded code running in a Java sandbox.

However, the security implications of downloading Java, PostScript, ActiveX, or Microsoft Word programs from arbitrary Web pages are significant, since such downloaded applets could easily contain malicious code, such as trap doors, Trojan horses, or viruses. Such malicious code could be downloaded and executed by a simple click on a Web hypertext link, yet the innocent user might not even know that he or she was downloading a program. Similar attacks are also possible from MIME attachments to electronic mail.

The designers of Java were aware of such issues, and built a number of features into Java to reduce the risks of downloaded applets. However, analysis by a team from Princeton University [2], as well as by a number of others on the Internet, has shown that the existing countermeasures in Java have weaknesses. It is very important to note that Java security has received a great deal of attention, precisely because the Java designers attempted to solve the problems. No one has attempted to solve these same problems for the other languages.

Limiting the damage potential of malicious applications is perhaps the hardest problem in all of computer security. It is the reason that the computer security community developed the concepts of a Trusted Computing Base (TCB), lattice security models to enforce confinement on untrusted applications, and high levels of assurance to avoid the problems of exploitable implementation flaws. Unfortunately, in the rush to support downloading applications from the Web, many of these computer security principles were overlooked by the developers of recent Web technologies.

To succeed against the highly sophisticated attackers that we see on the Internet today, the Java sandbox needs underlying operating system support to isolate applets from each other and to ensure that any given applet only gets access to exactly the information that it needs to perform its task and nothing else. Such operating system security support is unavailable in the most widely used client systems, such as DOS, Windows 95, the Macintosh OS or OS/2. Systems based on UNIX or Windows NT provide at least some assistance, because they support a separate user and supervisor state, file access controls, and can limit the damage a user process can do. To take advantage of such a system, the Java Virtual Machine (JVM) would have to be modified to run each applet in a separate process or address space. Even stronger protection could be afforded by a capability-based system, such as OS/400, to limit the access rights of an applet to exactly the information needed and no more. Similar techniques could be used for the other languages used for downloaded applications.

To allow customized access rights, each downloaded program needs to be digitally signed to unambiguously identify its source and to allow a decision to be made of what rights to grant the downloaded program. However, digitally signing downloaded programs without the corresponding operating system support provides only very limited benefits, because one downloaded program could attack another downloaded program and steal its privileges.

The level of sophistication of the attackers on Internet has significantly grown in recent years. This has been exacerbated by the spread of attack toolkits in the underground to allow relatively unsophisticated attackers to carry out very complex attacks that they could not have implemented by themselves. The types of attack commonly seen today on the Internet are as bad as anything envisioned by the original authors of the Orange Book as needing B3 or A1 levels of security. The days of commercial users only needing C2 are long past. Downloaded hostile applications from the Web can only be controlled by applying systems of that high a level of assurance. Unfortunately, such high assurance systems are still not generally available nor will they be in any near timeframe.

In summary, IBM is strongly committed to Java technology. We believe it offers many benefits in the implementation of platform-independent Internet applications, and we will offer Java in many of our products. However, IBM is also aware of the security risks when Java applets are downloaded from the Internet. These risks are not unique to Java, but are also present in ActiveX, Postscript, Microsoft Word macros, and many other languages. We want to offer our customers both guidance and product features to use Java technology wisely and securely.

1. J. P. Anderson, Computer Security Technology Planning Study,
ESD-TR-73-51, Vol. II, HQ Electronic Systems Division, Hanscom AFB, MA, October 1972, pp. 58-69.

2. D. Dean, E.W. Felten, and D.S. Wallach, Java Security: From HotJava to Netscape and Beyond,
Proceedings of the 1996 IEEE Symposium on Security and Privacy, May 6-8, 1996, Oakland, CA, pp. 190-200.

Webware: Widely Distributed Computation Coming of Age

James A. Roskind
Netscape Communications Corporation
Mountain View, California

Webware provides both a blessing and a curse, as applications can be written, and distributed to thousands, if not millions, of client sites via the web. The blessing comes with the immense computational capability, and scalability that results (you get an extra CPU for each client that participates). The curse comes as malicious, and poorly written applications become little more than a “click away” from arriving and running on these systems. In the end, this curse must be viewed in context, where we are running on insecure operating systems, and using many security-problematic applications. The most encouraging point with regard to this curse is the amount of attention security is getting. Although history has shown that this curse involves a large battle, we're hopeful that the intense effort being directed at webware will result in reasonable solutions.

The Java Language, and more significantly the designers and implementers of Java execution environments, are working to help guard the safety of all those clients. As pointed out by others on the panel, this guarding activity is not a new job, but the immensely widespread usage of the Web has suddenly brought millions of surfers into close proximity with potential attackers. The question this panel seems to be concerned with is how achievable is this task of guarding the clients?

There are other non-Java related questions that can also be considered when discussing “webware.” Most of these other approaches to webware security use a “binary trust” (i.e., trust is all or nothing) model. This alternative approach is modeled after the "shrink wrap model" of software security (trusting either the distributor, or the manufacturer). This trust model is part of what has caused the MS/DOS/Windows platform to grow to cover the globe at a nearly unbelievable rate. Adopting this “binary” (all or nothing) trust model also has caused a whole new industry to emerge, in the form of “virus scanners.” I would add that these scanners seem to find the greatest utility when a user misplaces trust in a distributor (example: trusting a BBS). In comparison with BBS’s, I think there have been significantly fewer distributions of malicious code directly from software manufacturers (the most common such distribution is a viral tainted master disk that goes undetected). In cases where trusted vendors have not directly inserted malicious code, there have been examples where these vendors put insufficient effort to prevent abuse of their applications (example: MS Word Macro Virus). In the end, shrink wrap is only as good as the manufacturer, and the binary trust model forces a user to place ultimate trust in software that neither needs, nor deserves such status.

Unfortunately, even on non-Java systems where efforts to provide more discrete levels of trust (example: Unix, where root and non-root processes exist), the massive interconnect of the Internet has brought forth rapid dissemination of the notorious Morris Worm program. Hence it is clear that even with some multi-level security, there is no easy solution. It is fear, awareness, and concern that is driving much of the security development in Java.

The Java security model starts with a remarkable premise (goal?): It is possible to restrict Applets to exist and run in a "sandbox." This sandbox is expected to be so restrictive, that even the most malicious of applet cannot proceed beyond clearly labeled bounds (example: can't modify the local disk; can't make local network connections that would be denied to the machine sourcing the applet; etc.). The first question that probably needs to be considered is whether it is possible to allow useful programs to run, and yet constrain them to a sandbox. To date, we appear to have found implementation errors in specific systems, but the architected approach seems to be working. It is hoped that the widespread dissemination of the Java source code by JavaSoft will accelerate the public scrutiny of the implementations, and that the sandbox model will not be the weakest link in a system-wide security chain.

The second layer of the Java security model will involve signed applets. This layer builds upon a cryptographically ensured channel between the manufacturers and the users. Signed applets will, based on a user's (or administrator's) decision, be granted access to potentially larger and larger sandboxes (some of which are clearly as permissive as arbitrary native code execution). As with the "binary trust" model described above (or the underlying shrink wrap model), the skill, concern, and security savvy of the manufacturer quickly becomes the critical question. Considering the context of running a Java applet on a platform or operating system which is supplied with only traditional shrink wrap assurances, it appears that this second layer is at least as strong as its foundation of support. The fact that manufacturers must consciously sign applications will raise awareness of these security issues.

One argument that could be made about weaknesses of permitting Java to trust many different manufactures (at the user's request), is that eventually the users will get sloppy, and an untrustworthy applet will become empowered. It could be argued that there are fewer OS vendors for a user to make a "sloppy" security decision about, and that users are more informed about the trustability of those fewer vendors. In contrast, there will soon be thousands of instantly accessible applets from thousands of manufacturers (cross-platform portability of Java accentuating the "problem" of instant web accessibility). Java hopes to address this growth by allowing both restricted sandboxes (various sizes), and providing facilities for centralized administration (critical to allowing companies to centrally restrict software use to certified "safe" applications). Java is bringing a great deal of power to webware, and is working to provide tools to harness that power (safely).

As mentioned earlier, the context must always be considered when discussing webware. Webware runs on platforms and operating systems having dubious security credentials. It can typically be expected that these support services will often be the weakest links in the security chain. Considering the wide-spread scrutiny that Java source code has received, and the lack of scrutiny for most operating system code, it becomes very believable that Java will soon be far from the weakest link in the security chain. Other webware without a trust model (BBS/public domain software), and even webware supported only by a binary trust model, will rapidly be seen as more problematic than Java. Hopefully these nightmares will not prevent the dream from being realized, as Java and similarly scrutinized languages arrive and are developed.

DARPA Research Panel 1: Secure Systems and Access Control

Panel Chair: Teresa F. Lunt, DARPA

Panelists:
Dan Sterne, TIS
Roshan Thomas, ORA
Mary Ellen Zurko, OSF
Jay Lepreau, University of Utah
John Rushby, SRI International

Over the past two decades, much of the research in computer security has been sponsored by the DoD and has focused on multilevel security (MLS). Several laboratory prototypes were built to demonstrate the feasibility of high-assurance MLS systems. However, very little of this work has transitioned. While many vendors did produce MLS versions of their products, these generally diverged from the standard products. This divergence leads users to prefer the standard versions, since most of the popular applications will not be available or may not work correctly on the lesser-known MLS versions. And this, in turn, means that those users who need MLS still do not have an affordable solution; much customization and special-purpose applications and integration code must be written.

Instead, what is desired is for vendors to build security into their mainstream products. This is feasible only if a large segment of users want the security. The security features of general-purpose products must meet the needs of a broad set of interests, not just MLS. Policy-neutral security mechanisms could enforce any number of organization-specific policies, including MLS, but would not have any single policy "wired in." These mechanisms should allow a broad enough set of policies to be specified and enforced so as to appeal to a wide set of user communities, such as finance, health care, and commerce, as well as defense. One can envision a future in which national-security-"blessed" policies will be available from third-party vendors for use with these generic, but specializable, products.

Most organizations have more complicated information protection needs that simple mandatory and discretionary access control matrix-oriented policies are capable of expressing. In addition to the familiar mandatory and discretionary access control policies, we should also explore richer policies such as role-based, task-based, and workflow-based policies so as to appeal to the broadest possible constituency. To do this, we need to identify a desirable range or class of policies, investigate natural ways of expressing such policies, identify and develop a common set of mechanisms capable of enforcing the desired range of policies, develop policy "compilers" to map user-specified policies into the base mechanisms, and address the related assurance issues.

Based on current research trends in operating systems, we expect future systems to be more modular. This may also be true someday of database systems. This will give us the opportunity to make security a modular and reusable component of systems. This has the advantage that the end user need only use the security modules if they need and are willing to pay for the security. It also means that various degrees of security can be made available for use with the same products. Moreover, it may be possible for several different systems to share the same security "modules," so that a common security policy can be enforced across diverse system components. There is the additional advantage that security modules can be replaced by high-assurance national-policy-enforcing modules when the systems are used in certain defense applications.

The panelists explore these and other issues being investigated in the DARPA research program.

Domain and Type Enforcement Firewalls
Dan Sterne, TIS

The pervasive need for E-mail and world wide web services and the growing importance of electronic commerce have driven many organizations to connect their local area networks (LANs) to the Internet in spite of the significant security risks this incurs. As a defense, many organizations use firewalls to constrain interactions with the Internet, allowing only the use of those services and protocols deemed relatively safe. While firewalls are a valuable tool, they reduce but do not eliminate Internet security risks. For example, a firewall that permits outgoing E-mail cannot tell whether such E-mail contains the announcement of a company picnic or the minutes of a highly proprietary corporate strategy session. Consequently, it allows either to flow out to the Internet, indiscriminantly. Similarly, a firewall that permits LAN users to surf and view anonymous remote web sites freely will not protect LAN hosts from attack by malicious web pages containing executable content, e.g., postscript or Java.

Addressing these Internet security problems requires protection mechanisms beyond those provided by firewalls, namely, operating system security mechanisms. Unfortunately, mainstream UNIX systems (and other mainstream operating systems) provide only weak, discretionary protection mechanisms that are insufficient for these purposes. In addition, UNIX systems are relatively easy to penetrate. In part, this is because they are difficult to configure securely, even by expert administrators. Moreover, they rely on a large number of complex programs that execute with root privilege; an attacker that subverts a single root program gains control over an entire UNIX system. Multilevel secure operating systems provide stronger protection but are viewed by many organizations as inflexible and ill-suited to the security problems of the commercial world.

Under DARPA funding, TIS is developing an integrated approach for Internet security that combines both firewall and secure operating system technologies. The foundation of this approach is a previously developed UNIX prototype whose kernel provides Domain and Type Enforcement (DTE), an extended version of Bobert and Kain's Type Enforcement. DTE is a strong yet flexible form of access control that can be configured to support a variety of site-specific security policies. An administrator configures a DTE system by writing high level access control rules in DTEL, human-friendly, machine-interpretable policy language. DTE controls access not only to files and devices, but network communications services. In order for a process on a DTE system to be able to send or receive network traffic, the traffic must be labeled with a type that is specified in the DTE policy as sendable or receivable in the process's domain. The DTE prototype currently uses the option space in IP headers to store type labels and other DTE security attributes.

The other central component in this approach is a firewall that integrates DTE and the TIS Firewall Toolkit. DTE is used in the firewall in two ways. First, the firewall is made stronger by organizing the firewall operating system components and firewall application proxies into small DTE-enforced execution domains. This increases the firewall's resistance to penetration by an attacker. Second, the firewall is made smarter by incorporating into it cognizance of the DTE capabilities and DTE policies of hosts on the LAN it protects.

In this approach, the DTE firewall's role is to support local hosts' DTE policies and to coordinate its actions, including policy updates, with other DTE firewalls. These notions are being investigated via three phases of prototyping. In the first phase, a DTE firewall attaches DTE attributes to inbound traffic from the Internet and checks the appropriateness of labeled outbound traffic. It also selectively channels to DTE hosts important but potentially dangerous network services (e.g., Java) that may convey too much security risk for ordinary (i.e., non-DTE) hosts that are also present on the LAN. In the second, two distinct enclaves protected by DTE firewalls exchange cryptographically protected network traffic. This traffic includes DTE security attributes having semantics that have been mutually agreed upon by both enclaves. This allows role-based and other kinds of security policies supported by DTE to extend across the Internet to enclaves operated by different organizations. In this phase, the DTE policies of the enclaves protected by the DTE firewalls will differ but overlap. The policy overlap, specified in DTEL, defines the kinds of information that both enclave's owners have agreed to exchange. In the third prototype, Domain and Type Authority (DTA) Servers will provide directory-like network services so that firewalls can dynamically discover the types of information that can be exchanged safely with other firewalls.

The increased reliance of commercial and government sectors on the Internet and its associated technologies intensifies the need for improved Internet security. While firewalls and secure operating systems have critical roles to play, a comprehensive approach requires both. By combining these technologies synergistically, we hope to better address the growing security needs of the government and commercial sectors and enable the safe exchange of a broader array of services over the Internet.

Task-based Authorizations: A Research Project in Next-generation Active Security Models
Roshan Thomas, ORA

In this project, we develop a new paradigm for access control and security models, called task-based authorizations. TBA is particularly suited for emerging models of computing. In particular, this includes distributed computing and information processing activities with multiple points of access, control, and decision making. TBA articulates security issues at the application and enterprise level. As such, it takes a "task-oriented" or "transaction-oriented" perspective rather than the traditional subject-object one. Access mediation now involves authorizations at various points during the completion of tasks in accordance with some application logic. In contrast, the subject-object view typically divorces access mediation from the larger context in which a subject performs an operation on an object. By taking a task-oriented view of access control and authorizations, TBA lays the foundation for research into a new breed of "active" security models.

In a task-based approach to security, the basic entities are:

- Tasks and sub-tasks: these represent strands of activity.
- Authorizations: these are approval steps that occur at one more points in the lifetime of various tasks and sub-tasks.
- Dependencies: these are relations between authorizations and their encompassing tasks.
- Authorization policies: these are authorizations and dependencies combined to form meaningful expressions of authorization policies.

Central to the TBA approach is the notion of an authorization-step, representing a primitive authorization act. In the paper-based forms environment, the analog of an authorization-step would be an approval on a form, identified by a signature. The active aspects of the model can be attributed to the fact that TBA recognizes the interaction of authorizations and permissions as it occurs within the lifetimes of tasks and activities, thereby enabling it to take an active role in the management of authorizations and corresponding permissions.

The key research directions that we are investigating during the course of this project include the following

- TBA as an active security model
- modeling and specification of authorization policies
- use of visual languages to specify authorization requirements and policies
- application of TBA to distributed computing and workflows

A model such as TBA can be used to address the gap that exists today between the enterprise and systems perspectives of security. Thus TBA can form a bridge between high-level enterprise security models and low-level access control models. TBA will have broad applicability in areas such as the automation of mission critical command and control scenarios where authorization sequences need to be carefully controlled, security management of complex operations in high-assurance client-server environments, as well as in forms-based workflow applications such as logistics management, distributed planning and claims processing.

User-centered Security and Adage
Mary Ellen Zurko, OSF

While "user-friendly security" is viewed as a humorous oxymoron in some circles, the security community has long acknowledged the importance of usable secure systems. There was a pragmatic recognition that secure systems that are difficult to use will get circumvented or insecurely managed by their users. In 1975, Saltzer and Schroeder identified psychological acceptability as one of eight design principles for computer protection mechanisms [2]. While other principles from that paper such as least privilege and fail-safe defaults have become standards in the security literature, there has been very little work done on user-friendly security. The lack of work in this area is due in part to the history of research, development, and use of secure systems. Most research and development in secure systems has strong roots in the military. People in the military are selected and trained to follow rules and procedures precisely, no matter how onerous. This user training and selection decreased the pressure on early secure systems to be user friendly. In another example of military influence, the first security model to achieve widespread attention in the security literature (Bell and LaPadula [1]) encoded military classification levels and need-to-know categories.

Much effort was then spent trying to apply this model to all uses of secure systems. Finally, mathematical rigor has been emphasized over usability in many security modeling efforts.

In considering how best to integrate usability and security, we considered three different approaches. We can apply established procedures for enhancing usability to developing or existing secure systems.While this approach seems the most obvious and the cheapest, it has rarely been documented. A second approach is to integrate appropriate security services into software with a strong usability component, such as mass-market applications or groupware. Most of the work in this area has focused on privacy, and has taken place in the Computer Human Interface (CHI) community. We call the third approach user-centered security[4]. The term refers to security models, mechanisms, systems, and software that have usability as a primary motivation or goal. This approach provides the tightest integration between usability and security.The timing seems right for a renewal of interest in synthesizing usability and security. There is increasing pressure on government funded researchers to produce results that can be used to solve real world problem, and the standard for ease-of-use in commercial products continues to rise.

We are pursuing our vision of user-centered security in the Adage project (Authorization for Distributed Applications and Groups) [3]. Adage will provide a toolkit that will allow distributed applications to take advantage of Adage's services, encouraging consistent mechanisms and policies throughout an organization. Adage is specifically conceived to overcome the usability problems with authorization mechanisms for distributed applications in use today.

The first of these usability problems is that the applications unnecessarily export the underlying data structure as the user model. The user metaphor for Access Control Lists (ACLs) is the ACL data structure; for system masks it is the system mask. The user is given a rudimentary formatted display of the information in the data structure (or perhaps just a literal display of its values) and must learn the algorithm that the computer software will use to evaluate that data structure in order to understand what access control policy is actually instantiated. A large gap remains between these traditional security mechanisms and a user's or site's security policy, stated in natural language. By analogy, ACLs are the assembly language of security policy. They are a complex, low-level language. Only an expert in a particular implementation of ACLs can hope to program it correctly the first time. ACLs have the added disadvantage of being difficult to test without making changes on a live system. One component of Adage will be a higher-level authorization language that begins to close the gap between security mechanisms and site security policies. It will come with a visual builder that allows site security administrators to build up an authorization policy from visible policy pieces. Furthermore, these policies can be shared with other domains. The primitives supported by this language will support a wide range of user and application policies, because they will be based on security policies actually in use and on interviews with security administrators.

One insight that Adage shares with current work on roles is that within organizations it is natural to think about both users and objects in terms of how they relate to each other and what place they fill within the organizational structure. Adage will use groupings to reflect these intuitions. It will use groupings of objects and of actions to more easily refer to objects and actions in a security policy. Groups of users and their roles will receive particular attention. Adage will provide an infrastructure for defining the relationships and restrictions on groups and roles that will allow it to support models from both the security and groupware literature. For example, two groups can be restricted to have no membership overlap, to support static separation of duty. Users taking on the role of Chair can be restricted to those users in a particular group.

Adage will continue the work in user-centered trust models by modeling common trust dimensions such as amount of trust (How much do I trust you? How much do I distrust you?) and type of trust (What do I trust you for?). Adage will apply this trust model to services whose information is used as input to authorization decisions (such as authentication servers and group membership servers). This will allow an enterprise to articulate a trust policy and have it apply to all its authorization decisions. In addition, the model will allow trusted services to introduce other trusted services, forming chains of trust where the amount of trust degrades over hops, much as real-life trust does.

[1] Bell, D. E. and L. J. LaPadula. Secure Computer Systems: Unified Exposition and Multics, Technical Report ESD-TR-75-306, The MITRE Corp., March 1976.

[2] Saltzer, Jerome H. and Michael D. Schroeder. "The Protection of Information in Computer Systems", in Proceedings of the IEEE, 63(9), 1975.

[3] Zurko, Mary Ellen. Adage home page, http://www.osf.org/www/adage/index.html.

[4] Zurko, Mary Ellen and Rich Simon. "User-Centered Security", in Proceedings of New Security Paradigms Workshop, 1996.

Encapsulated Environments Using the Flux Operating System

Jay Lepreau, University of Utah

Most modern operating systems provide a concept of "virtual machines" — e.g., processes or tasks — and allow several such virtual machines to coexist on a single machine and compete with each other for hardware resources. Such separate processes are a classic way to support separate information domains. In the 1970's the term "virtual machine" usually referred to an OS architecture that exported what appeared to be the naked hardware, and an entirely separate copy of a stand-alone operating system ran on that "virtual machine."

Based on a synthesis of microkernel and virtual machine concepts, we have developed an OS architecture that allows recursive virtual machines (virtual machines running on other virtual machines) to be efficiently implemented, in software, by a microkernel running on generic hardware. The model can also be called a "nested process model," in which *any* process can completely contain and control other processes within it.

Virtual machines were a classic way to provide high security subsystems, fully isolated from one another. Our recursive model takes this a step further, efficiently providing *hierarchical* control by any process in the system. Such flexible and hierarchical control is ideally suited to supporting the security requirements of arbitrary untrusted applications, often loaded over the Internet and Web. Each security manager can completely control the resource (memory, cpu, higher-level services) of its children. Each child may, if it wants, implement similar control over its children. In this manner the children can control and isolate further untrusted applications.

DTE Firewalls

Dan Sterne
sterne@tis.com

Lee Badger
badger@tis.com

Trusted Information Systems, Inc.
3060 Washington Road (Rt. 97)
Glenwood, MD 21738

http://www.tis.com/docs/Research/dtefw.html
http://www.tis.com/docs/Research/DTE.html

Problem

Many organizations are connecting to Internet in spite of security risks

Firewalls help, but are not enough:

- too many services must be restricted (e.g., NFS, X11)
- security perimeter is inflexible
- no protection of sensitive data
- no protection from inside attacks
- limited protection from content-based attacks (e.g., Java)

Need supporting security from operating system (OS), but ...

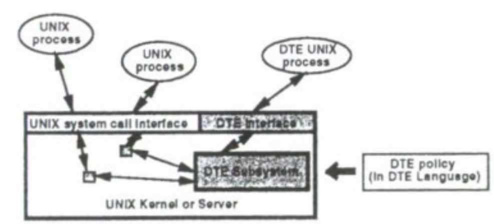
- mainstream OSs (e.g. UNIX) provide only weak, discretionary mechanisms
- MLS OSs strong but inflexible

Solution Strategy

Combine three technologies:

- Internet Firewalls - regulate and filter services
- Domain and Type Enforcement (DTE) - secure UNIX
- Cryptography - protect communications over Internet

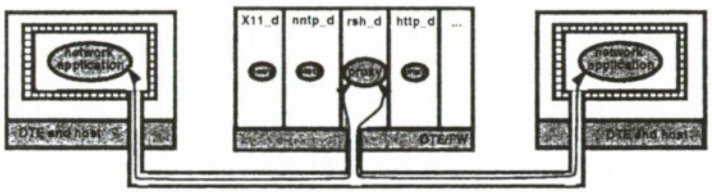
Domain and Type Enforcement (DTE) *



- Strong, flexible access control for operating systems
- Security policies specified in high-level DTE Language (DTEL)
- Backward compatible with UNIX programs, TCP/IP networks.
- Labels and mediates network messages.

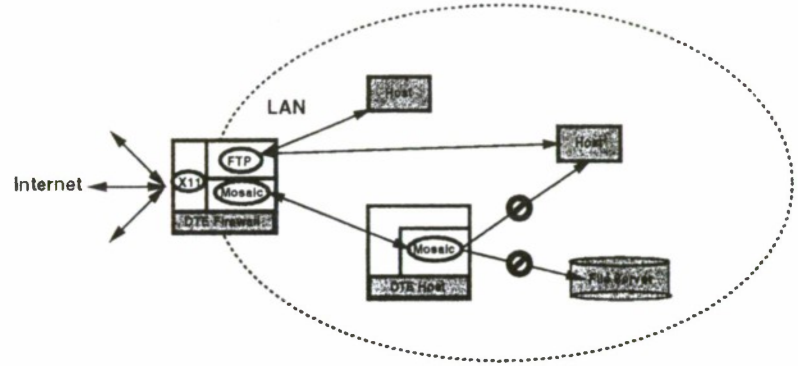
* An extension of Boebert and Kain's Type Enforcement

DTE Firewall Strategy



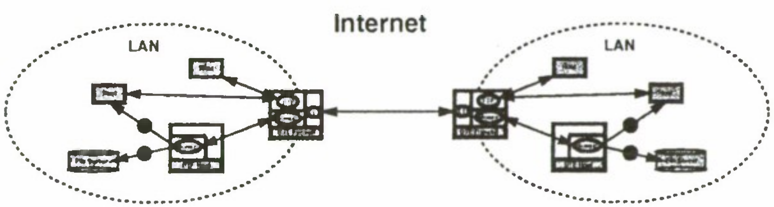
- DTE hosts confine applications
- DTE firewalls:
  - coordinate DTE policies between DTE hosts
  - associate DTE attributes with data from non-DTE hosts
  - confine network proxies

Phase 1: DTE Firewalls



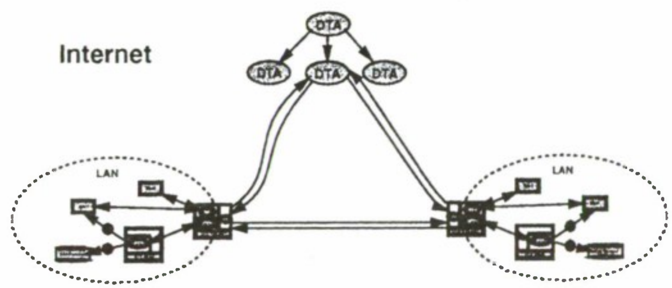
- Safely use more network services
- Stronger firewalls
- Encapsulated network processes
- Protects sensitive information

Phase 2: Distributed DTE Firewalls



- Encryption between Firewalls
- Restricted environments span LANs
- Coordinated protection via DTEL

Phase 3: Domain and Type Authority (DTA)



- Establishes trust relationships
- Provides authentication
- Distributes DTEL modules
- Dynamic policy discovery service

Task-based Authorization:

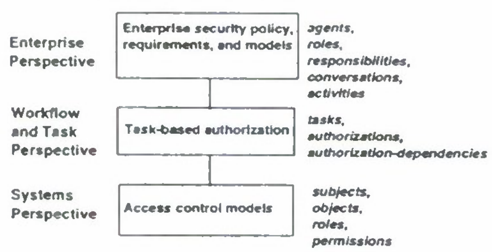
A Research Project in Next-generation Active Security Models

http://www.oracorp.com/tba

Roshan Thomas
Odyssey Research Associates
301 Dates Drive
Ithaca, NY 14850
rthomas@oracorp.com

1

TBA as bridge between enterprise and systems security



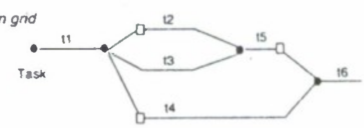
2

Task-based versus Subject-object Access Control

subject-object access matrix

Subjects \ Objects			
USER-A	R	R, W	
USER-B	W	R	

taskand authorization grid



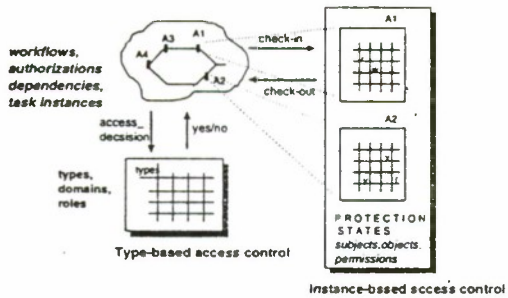
3

Subject-object vs task-based

- **Subject-object access control**
  - data structure view of security information
  - unrelated units of security information
  - no memory of evolving context
  - subject-object models are passive
- **Task-based access control**
  - fundamental abstractions are tasks and authorizations
  - view security for tasks and policies
  - authorizations are related through dependencies and scope
  - active management of authorizations

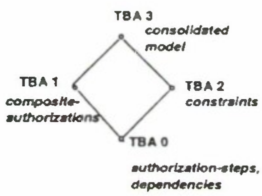
4

TBA as an active security model



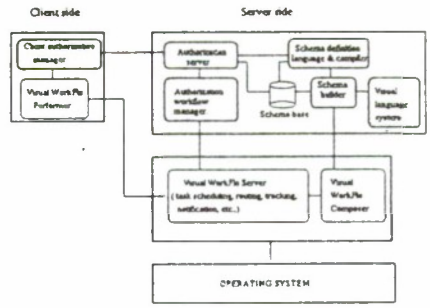
5

A framework for a family of TBA models



6

Prototype Architecture



7

Impact

- Secure management of mission-critical, logistics, and workflow applications
- Distributed systems for authorization management
- Advanced enterprise security tools
- Promote greater awareness for next-generation security models

8

Adage:

Authorization for Distributed Applications and Groups

Mary Ellen Zurko
OSF Research Institute
zurko@osf.org
http://www.osf.org/~zurko/
http://www.osf.org/www/adage/

Adage Motivation

Current Problems

Complex administration

- Tools are low-level (ACLs)
- Poor tools for grouping large numbers of subjects/objects

Inconsistent mechanisms across applications

Limited notions of distributed trust

- No tools for high-level policy statement
- Only highly structured or anarchic trust infrastructure and only for authentication

Adage Context

Secure Distributed Authorization

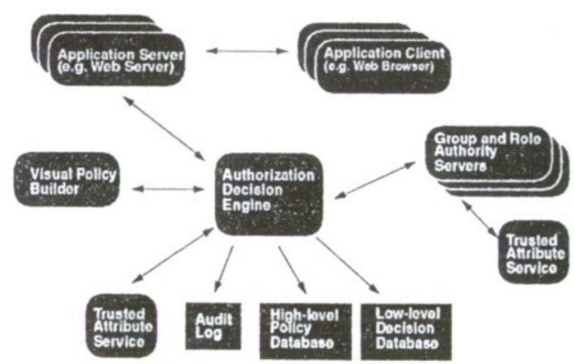
Goals:

- Emphasis on communication within a single geographically distributed organization
- Policy-neutral: Applicable to multiple environments
- "User-friendly security" is not an oxymoron

Non-goals:

- Integrity, privacy, or authentication research
- Applicability to global, unmanaged environments

Adage Architecture



Authorization Tools

Visual Policy Builder for High Level Authorization Language

Inputs include

- user and object attributes (names, groups, roles, labels, ownership)
- contextual information (transaction history, time)

High-level Security Policy Database

- Platform-independent policy representation
- Sharing of policy primitives between organizations

Group and Role Authority Server (GRAS)

Support for groups and roles with rich semantics, including relationships and restrictions between them

Types of groups and roles derived from models and policies in the security and groupware literature

Multiple group authorities hold different group memberships for a single authenticated identity

A single enforcement engine manages authorization and group information

Trust Model

Framework for users to think about and use authorization for organizational policies

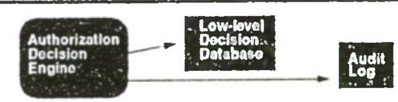
Underlying model provides a consistent foundation for common trust dimensions

- Supports notions of amount and kind of trust and trusted referrals

Trust model matches user expectations about how security should work

- Security, performance and usability trade-offs

Enforcement Engine



Underlies all of the Adage components

Low-level Authorization Decision Database

- Platform-dependent policy representation
- Contains low level representation of authorization data, such as ACLs
- Performance sensitive to application needs

Auditing support

Encapsulated Environments using the Flux Operating System

Jay Lepreau

Department of Computer Science
University of Utah

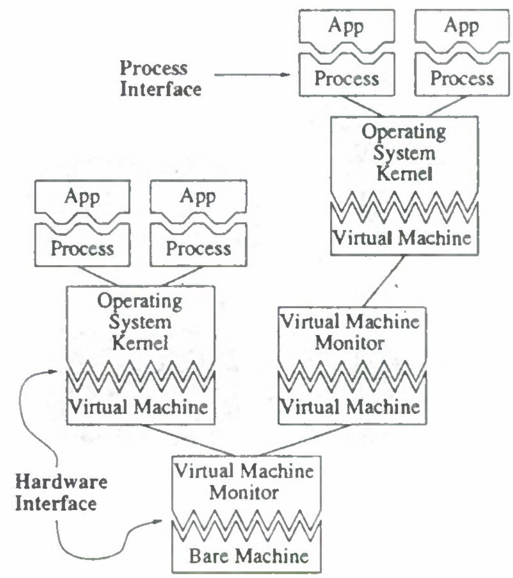
http://www.cs.utah.edu/projects/flux/

lepreau@cs.utah.edu
801-581-4285

National Information Systems Security Conference
October 1996

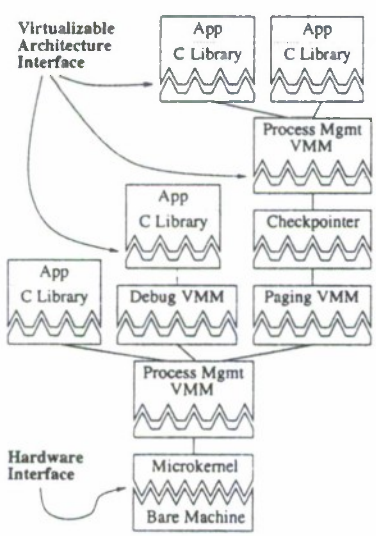
1

Classic Virtual Machines Based on Hardware Architectures



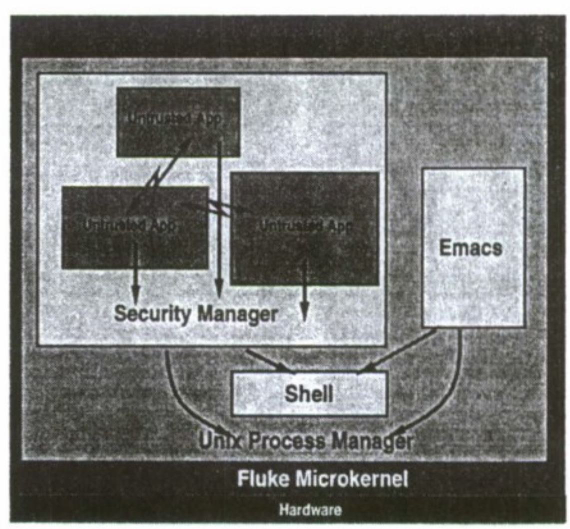
2

Flux-style Virtual Machines Based on Software Architectures



3

Secure Environments for Untrusted Applications using Flux Virtual Machines



4

Virtual Machines Provide Isolation

Strong and *flexible* isolation

- Between arbitrary subsystems
- Addresses denial-of-service
- Addresses covert channel control

5

Virtual Machines Provide Resource Accounting and Control

Strong and *flexible* control of:

- Memory
- CPU
- Higher-level services

6

Flux OS Status

- Running on PCs, supports POSIX subset
- Several virtual machine monitors running
- Small kernel, layered implementation
- Kernel API and design document published
- NCSC INFOSEC R23 collaboration

7

Conclusion

Flux recursive virtual machines provide:

- Isolation
- Resource accounting and control
- Stackable encapsulated environments and security monitors

8

Panel

INFOSEC Research and Technology

Facing the Challenge: Secure Network Technology for the 21st Century

Panel Chair: Mr. Richard Schaeffer, Office of INFOSEC Research and Technology, NSA

Panel Members:

Mr. Bob Meushaw, NSA, Technical Director
Ms. Chris McBride, NSA, Technical Staff
Mr. Dave Muzzy, NSA, INFOSEC Cryptology
Dr. Lee Taylor, NSA, INFOSEC Engineering
Dr. Blaine Burnham, NSA, INFOSEC Computer Science

INTRODUCTION

The Office of INFOSEC Research and Technology is focusing its efforts on the network security challenge. It is integrating core expertise in the areas of INFOSEC Cryptology, INFOSEC Engineering, and INFOSEC Computer Science and focusing these disciplines against a broad spectrum of INFOSEC research initiatives. In addition, the Office has engaged in a number of activities in order to consolidate, focus, and better leverage its research investment. These activities include:

* Development of the INFOSEC Research and Technology Program, a means of coordinating and collaborating with the INFOSEC research community in government, industry, and academia.
* Participation in the Joint Technology Office, JTO, a joint initiative between DARPA, DISA, and NSA.
* Development of a Technology Forecasting process to support the identification of major research investment areas.

PANEL DISCUSSION

The panel will present an overview of the integrated approach that the Office of INFOSEC Research and Technology is taking to address the network security challenge. The panel will expand on each of the initiatives mentioned above and discuss how these activities will result in a more effective INFOSEC research program.

DEMONSTRATIONS

The Office of INFOSEC Research and Technology will be demonstrating a subset of their suite of core technologies to include Token Technology, Voice Verification, Real Time Encrypted Voice, Security Services Applications Program Interface, Firewalls, Secure Wireless Communications, and Tamper Protection Display, and Assurance Metrics.

Toward a Common Framework for Role-Based Access Control

Panel Chair:
David Ferraiolo, National Institute of Standards and Technology

Panelists:
1. Dr. Ravi Sandhu, George Mason University
2. Dr. Virgil Gligor, University of Maryland
3. Rick Kuhn, National Institute of Standards and Technology
4. Thomas Parenty, Sybase

Introduction

Role based access control has been used in computer systems for at least 20 years, but only within the past few years have rigorously defined general purpose Role-based Access Control (RBAC) models begun to appear. Lately, there has been great interest in RBAC. RBAC has captured the attention of major vendors and researchers. For instance RBAC properties are now being directly designed into database products and several articles from around the world immerged. To maintain this momentum and to allow RBAC to reach its full potential, we must approach RBAC from the perspective of enterprise computing in the commercial arena. In other words, how will RBAC help in providing cost-effective information technology solutions to carry out the business activities of enterprises? The recent flurry of activity in RBAC suggests that RBAC has the capability to serve security requirements that are not being met by currently available systems. The purpose of this panel is to discuss the current state of RBAC research and future directions in research and implementation of RBAC.

A role is chiefly a semantic construct forming the basis of access control policy. With RBAC, system administrators create roles according to the job functions performed in an enterprise, granting permission (access authorization) to those roles, then assigning users to the roles on the basis of their specific job responsibilities and qualifications. The benefits to an enterprise are ability to administratively specify and enforce enterprise specific security policies that can not be achieved using other methods of access control, and to dramatically streamline the typically burdensome process of authorization management.

Why are roles special?

The central notion of role-based access control is that users do not have discretionary access to enterprise objects, but instead access permissions are administratively associated with roles, and users are administratively made members of appropriate roles. It has been felt that this idea can greatly simplify management of authorization data while providing opportunity for great flexibility in specifying and enforcing enterprize specific protection policies. Roles can be created for various job positions in an organization. Users can be made members of roles as determined by their responsibilities and qualifications, and can be easily reassigned from one role to another without modifying the underlying access control structures.

In some cases the potential benefits of RBAC have been accepted by both users and vendors, without a precise definition of what constitutes RBAC. In the past RBAC features have been implemented in enterprise applications, without a frame of reference as to its functional makeup, making RBAC an amorphous concept interpreted in different ways by users, researchers, and system developers.
There is a clear need to define and guide the evolution of a reference model for RBAC that is vendor neutral and mechanism independent and serve as a unifying force. From a commercial standpoint, we have to consider how RBAC fits into emerging models of computing, to include massive distribution such as internet, interoperable objects and software components, and workflow automation.

To promote the advancement and definition of RBAC the National Institute of Standards and Technology (NIST) is conducting and sponsoring research in the area of RBAC. To date three independently developed efforts on RBAC are underway at NIST: a Small Business Innovation Research (SBIR) program with Dr. Ravi Sandhu of George Mason University and Seta Corporation to help define RBAC and its feasibility, an effort with NSA's R23 Research and Engineering group and Dr. Virgil Gligor of the University of Maryland to create a formal model and implement RBAC on a policy-independent Mach microkernel-based operating system being developed by R23 called Synergy, and a Advanced Technology Program (ATP) effort being led by John Barkley of NIST to demonstrate how RBAC can be used for a health care system. As a result of these and other research efforts into RBAC, a number of well defined RBAC approaches and model have been created.

Common Model for RBAC

To date this RBAC research has yielded success in that advanced properties and models of RBAC are now widely available through numerous publications on the subject. In some cased viability of advanced RBAC features have been demonstrated through implementation and their application. There are even signs that some of the more advanced properties of RBAC are now being designed and implemented within significant and well established commercial products.

Although, the state of the technology has advance considerably over the past few years, there still does not exist a single or defacto standard for RBAC. Work is now being conducted to develop a consolidated model of RBAC that takes advantage of past and existing research. While there does exists a good amount of agreement as to what constitutes RBAC, many differences do exist.

An obvious question is whether there should be a common, widely accepted, model for RBAC, as there is for multi-level security. If so, what model should be used? It is probably too early for a formal standard for RBAC, but we are likely to see a common model begin to emerge as industry implements role based systems. One RBAC specification that has already been implemented in commercial systems is included in the latest SQL3 database standard. But many applications have requirements that differ from database systems, so a general purpose model for RBAC may look different from that defined for SQL. For example, many applications may require dynamic separation of duty, which is not part of the SQL3 definition of RBAC. Other open consensus specifications with RBAC components include the Secure European System for Applications in a Multi-vendor Environment (SESAME), and the RBAC example included in the Object Management Group's Common Object Request Broker Architecture (CORBA).

The motivation for this panel is to publicly describe and compare some of the more prominent RBAC approaches that exist today.

It is expected that the panel members with their diverse backgrounds will bring both an industrial and academic perspectives to the discussion.

Panel

MISSI SECURITY MANAGEMENT INFRASTRUCTURE

The Certificate Management Infrastructure:

Now and In the Next Year

Moderator

Alfred W. Arsenault

Technical Director, Network Security Management Division, NSA

Panel Members

Donald R. Heckman, NSA

Robin L. Gerretson, NSA

Steven Capps, NSA

Abstract

The Multilevel Security Information System Security Initiative (MISSI) is fielding solutions that provide individual identification and authentication, access controls, and other security services that together provided "writer-to-reader" security for network applications. MISSI relies on each entity in the system having a "certificate" that identifies the entity and its privileges, and provides the public key(s) which should be used when communicating securely with that individual.

Key to MISSI is the Security Management Infrastructure, which is responsible for the management of necessary security services. A major part of the Security Management Infrastructure is the Certificate Management Infrastructure, or CMI. The CMI is the set of equipment and people responsible for issuing, updating, renewing, rekeying, and revoking certficates.

The MISSI CMI consists of a Policy Approving Authority (PAA) at the top, with two Policy Creation Authorities (PCAs) below, a number of Certification Authorities (CAs) below that. The CAs are used to program FORTEZZA cards for MISSI users.

The MISSI CMI has been operational for approximately 18 months. In this session, we will review the status of the CMI, describe planned upgrades in the coming year, and describe how the CMI actually operates.

PANEL DISCUSSION:

This panel will begin by describing how the MISSI CMI came to be, and where it now stands. We will then move to a discussion of how the CMI is currently operating. We will conclude with a description of anticipated changes to the CMI in the next 12 months, including fielding support for commercial signature algorithms, providing certificates compliant with the Version 3 X.509 standards, and other changes.

Panel

Future of Trust in Commercial Operating Systems

Moderator
Todd Inskeep
Workstation Security Products Division
National Security Agency
tkinske@missi.ncsc.mil
(410) 859-4464

Panelists

Representatives from Vendors working with secure/trusted operating systems

Abstract:

Over the last 20 years, much effort has been spent on trying to develop technologies for highly assured, secure operating systems. There are many discussions that could question how far we've really come, but this panel will focus on where we are going. This panel will discuss how current trends in Information Systems, MISSI, DMS, COE, and the vendor products (Solaris, HP-UX, Windows NT, Trusted Mach) will affect the quest for higher trust in operating systems. The government's emphasis on commercial technologies, and commercialization of technology are likely to affect the push for Trust. At the same time commercial concerns about Internet security may be revitalizing the argument for higher trust. With this as a continuing trend, we'd like to discuss where assurance and functionality in commercial systems are going.

Vendors Experience with Security Evaluations

Panel Overview: This panel is composed of managers of security product evaluations from several US-based corporations which have extensive experience in undergoing TCSEC and ITSEC based evaluations for commercial-off-the-shelf (COTS) products. Panelists will present and contrast their experiences in achieving successful TCSEC and ITSEC evaluations, identifying challenges and successes in each of the respective processes that made each of their evaluations a "win-win" with respect to their market and the evaluation community. Panelists will present a set of "lessons learned", which will be useful for companies considering evaluation in order to sell their products in the US and international markets.

Chair

Jeff DeMello
Director, Worldwide Security Evaluations
Oracle Corporation
500 Oracle Parkway, Box 659410
Redwood Shores, CA 94065
+1.415.506.8797 *voice*
+1.415.506.7226 *fax*
jdemello@us.oracle.com

Panelists

Duncan Harris
ITSEC Evaluations Manager
Oracle Corporation
500 Oracle Parkway, Box 659410
Redwood Shores, CA 94065
+1.415.506.4007 *voice*
+1.415.506.7226 *fax*
djharris@us.oracle.com

Ken Moss
Program Manager
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399
(206) 936-7774 *voice*
(206) 936-7329 *fax*
kenmoss@microsoft.com

Ian Prickett
Security Evaluations Manager
Sun Microsystems
2550 Garcia Avenue, USJC01-104
Mountain View, CA 94043
(408) 953-4825 *voice*
(408) 428-9411 *fax*
ian.prickett@ebay.sun.com

Janice Caywood
Secure Unix Evaluation Project Manager
Digital Equipment Corporation
110 Spit Brook Road, ZK3-2/X74
Nashua, NH 03062
(603) 881-2919 *voice*
(603) 881-2379 *fax*
caywood@zk3.dec.com

Panelists Statements

Duncan Harris, *Oracle Corporation*

Introduction

Oracle Corporation successfully completed US TCSEC and UK ITSEC evaluations of the Oracle7™ and Trusted Oracle7™ database servers over two years ago. As layered products, these evaluations were conducted on Hewlett Packard and Sun Microsystems operating systems in the US and UK respectively. Since these first evaluations finished, Oracle has continued to support both evaluation processes and has provided input to the development of the new international Common Criteria and the UK ITSEC Scheme's Certificate Maintenance Scheme (CMS) which is the UK equivalent of the US Ratings Maintenance Program (RAMP).

Cost and Duration of Evaluations

Security evaluations, wherever they are conducted in the world, are very expensive. The direct cost of an ITSEC evaluation is more obvious as that money is paid directly to a Commercial Licensed Evaluation Facility (CLEF), whereas a US evaluation, though apparently free, costs much more in development staff time. Oracle published a paper just after the first evaluations which notes the cost of Oracle's US and UK evaluations to be $850,000 and $600,000 respectively. Since the UK evaluation, which started later, reaped the benefits of the US evaluation, Oracle estimates the nominal costs to be similar, though the US evaluation required much more development staff time and thus had a much higher opportunity cost. ITSEC evaluations are also of considerably shorter duration than their US counterparts (for Oracle, only 16 months of actual evaluation work in the UK vs. 42 months in the US).

Mutual Recognition

The ITSEC, although it was a more immature document, is now coming of age and is much more flexible in its approach, by, for example, allowing a vendor to specify a product's security functionality much more precisely than the strict, large granularity TCSEC classes. The US also now recognizes F-C2/E2 (or higher assurance) ITSEC certificates as equivalent to TCSEC C2. So, why evaluate a product in the US at all? Because TCSEC certificates are still widely recognized around the world, establish credibility in a vendor's security claims and help a vendor sell into otherwise sceptical markets. It is in fact difficult to put a precise value on a TCSEC or ITSEC certificate.

Re-evaluations (RAMP vs. CMS)

The regulations for the RAMP process are extremely onerous on vendors. Oracle has discussed the hefty workload with NSA and is pleased that a more flexible approach is being shown which should end up improving the quality of security analyses without any loss of assurance. By contrast the new UK ITSEC Scheme's CMS, although initially promising to be a simpler and quicker process than RAMP, appears to be going down the route that RAMP once took. The burden is placed back onto the vendor to employ one or more Developer Security Analyst (DSA) which is the UK equivalent of the Vendor Security Analyst (VSA) to undertake security analyses of all change to code. The CMS is likely to work well for small product vendors where the whole development team can be DSAs, but for vendors with large development groups the DSA tasks will probably be thrown onto the existing, dedicated, evaluation support team. This goes against the hope in the CMS that a DSA is actually a developer.

Evaluators' Experience

The experience of evaluators is invaluable. It does not go unnoticed that US evaluators tend to be people of many years of computing experience who are extremely well trained and knowledgeable, whereas many UK evaluators are trainees, often recent graduates. Although each CLEF has its fair share of experienced evaluators, the average UK evaluator's computing experience is undoubtedly lower than their US counterparts. The training of UK evaluators, although controlled and approved by the UK Certification Body, appears far less than in the US with much reliance on experience gained during evaluations. The commercial pressures on CLEFs do not help in this regard.

Therefore, when a vendor is considering which CLEF to choose for a first ITSEC evaluation, it is very important to insist on a choice of staff and to pick a balance of evaluators with solid ITSEC experience and with practical experience in use of the product to be evaluated (or a similar product). For an ITSEC re-evaluation, unless a vendor is particularly unhappy with the CLEF used for a first evaluation, it would be wise to use not only the same CLEF but to insist on the same evaluators. Invaluable training has, in effect, been poured into these people and that experience of the product's internals should not be ignored. A major expense that Oracle has endured from using multiple CLEFs is that of the considerable efforts of training evaluators on the complex internal architecture of its products. This cost is minimized by contracting to the same CLEF, as well as requiring the same evaluators, for new evaluations.

Inconsistencies between CLEFs

Although the ITSEC is one criteria, the ITSEM one methodology and the UK ITSEC Scheme's guidance to evaluators is the same for all CLEFs, it is surprising to find that considerably different approaches are taken to product evaluation by different CLEFs. All five CLEFs have varied reputations. Oracle has now had practical experience of three CLEFs first hand. Some CLEFs take a more formal approach and appear to act more as straight auditors, others are more informal but still carefully follow the UK ITSEC Scheme's rules. Before considering a first evaluation, or considering a change of CLEF, talk to other vendors about their experiences. And do not be swayed by CLEF's quoted evaluation costs alone - this figure is a relatively small percentage of the overall cost.

The new Trusted Technology Assessment Program (TTAP) scheme, which is destined to introduce CLEFs to the US, should be firmly controlled to ensure that the marked differences that are so evident between UK CLEFs do not occur in the US. Measures such as standard technical reports where deliverables' examination is consistently dealt with to the same depth should reduce the disparity in reporting found in the UK.

The Future: Common Criteria and Mutual Recognition

With the Common Criteria just around the corner, the critical issue for the future from vendors' perspectives is mutual recognition of certificates. Vendors' evaluation deliverables, the inputs for a Common Criteria evaluation, will become identical for the UK and US, so why should the certificates, the outputs, be different? Whether mutual recognition comes about soon or not, accreditors may soon take the attitude that a Common Criteria certificate is a Common Criteria certificate and ignore any subtle differences between national scheme rules. If mutual recognition doesn't come about quickly, vendors may vote with their feet and only evaluate against the Common Criteria in a country whose CLEFs are cheaper, quicker and more flexible. The new US CLEFs will have to be highly competitive to capture vendors' interest.

Summary

Evaluations have proved to be very expensive and time consuming. Some vendors have hesitated to start or continue evaluations for these very reasons. Oracle Corporation, however, has every intention of continuing to support our customers that have requirements for evaluated products. Oracle has a firm commitment to the evaluation community and the standards it promotes. We are running TCSEC and ITSEC evaluations right now, have plans for future TCSEC and ITSEC evaluations, and are looking forward to participating in the new Common Criteria evaluation process.

Ken Moss, *Microsoft*

Microsoft® Windows NT® Server and Windows NT Workstation operating systems have been evaluated against both the US TCSEC and the European ITSEC evaluation schemes. Although both processes provide an comparable level of confidence to the customer, the evaluation methodologies differ.

For a software engineering firm to build an operating system that will meet the needs of its customers and also be viable, it must satisfy numerous requirements. Examples might include scalability, performance, software developer support, device support, and usability features. These examples represent dynamic targets that must be constantly tracked for the longevity of a product line. Other requirements such as POSIX compliance, Orange Book security compliance, and Capability Maturity Modeling are more static requirements. These requirements are achieved through in depth analysis.

When considering the ITSEC scheme, the concept of an advocate based approach is highly appealing. This model allows a software engineering firm to do what they do best, which is to develop new product features, and leverages an ITSEC CLEF to focus on the evaluation. Who is better equipped to lead an evaluation than a CLEF that has driven the process several times before?

The strength of the TCSEC scheme focuses on the level of detailed analysis that is performed. The demand for information far exceeds those of the ITSEC requirements. Additionally, the TCSEC scheme provides a better mechanism for providing up front advice while designing a secure product.

The ITSEC scheme focuses more on the theory of design, testing and production and less on the actual design, testing and production.

The question becomes what measures are recognized by the customer. From a customer perspective security in general is misunderstood. For this reason most customers choose not to learn the technology and instead rely on the creditability of the independent evaluator. In this case, both evaluation schemes have proven to be creditable, but the US TCSEC process has shown greater commercial acceptance around the world.

The Common Criteria will benefit software engineering firms, primarily because the product is evaluated only once. The Common Criteria also will bring further confusion to the evaluation process. Few software customers today are aware of the Rainbow Series and the feature sets that comprise the security ratings. The Common Criteria will further perpetuate this problem as software venders and customers will then need a mastery knowledge of "profiles". It takes a only a few minutes to describe the difference between a C1 and C2 TCSEC rating, but where would one start when comparing an enterprise banking profile to a Wall Street trading profile? Which profile would you describe as containing a higher level of assurance? Which security target should be used in a given evaluation? Even if profiles existed that mapped one-for-one to the existing security ratings, the presents of "weaker" profiles dilute the credibility of the evaluation process. In other words, if every software product has a security rating and there is no clear way to compare the strength of a security profile, the evaluation process will have less substance.

Ian Prickett, *Sun Microsystems*

Security Evaluations At Sun Microsystems

Introduction

For the last year and a half I have been working for Sun managing its security evaluations. Initially I was focussed on the UK ITSEC scheme but I now have responsibility for managing Sun Federal's NCSC evaluation efforts as well. This has been a very challenging and rewarding role, during this time I have had to deal with a wide range of issues from a variety of sources including customers, systems integrators, internal management and development staff.

Evaluation Lessons

From this experience I have learned a number of lessons which I think will be of relevance to others considering whether to perform a security evaluation on their product I intend to cover these points in more detail in my presentation:

Evaluations are Expensive: NCSC Evaluations require significant resources to support correctly. ITSEC Evaluations require large amounts of money to be paid to external companies.

Evaluation Targets Evolve: Because of the timescales involved in major evaluations as well as the flexibility of the evaluation schemes, you should expect the criteria you have to meet to change over time.

You Should Evaluate What Your Customer Will Use: Evaluations are an exercise in risk reduction, you should always ensure that you evaluate your product as customers expect to use it i.e. if your customers expect to use client-server, current hardware, networking and window systems interfaces make sure you include them in your evaluation.

You Should Design The Product With Evaluation in Mind: It is significantly easier and cheaper to produce evaluation deliverable as you are producing the product that you intend to evaluate, rather than after the product has been completed.

Evaluations Are Global: Evaluations (especially ITSEC) are accepted by numerous countries world -wide, they are relevant to a huge marketplace not just the United States or Great Britain.

Evaluations are Relevant to Commerce as well as Governments: Commercial companies are beginning to see the relevance of recognised evaluation schemes, most of them are performing their own internal product evaluations already.

Evaluations Are Worthwhile: To sell into most government markets you have to evaluate your products, but the potential revenue far outweighs the costs of the evaluation itself.

Evaluation Successes

During this time we have had a number of evaluation successes at Sun, we have completed three ITSEC evaluations including the first full CMW (Trusted Solaris 1.2) to be evaluated by the UK ITSEC scheme.

Future Evaluation Issues

Looking to the future I am already working on the evaluations of the second generation of Sun secure products including products such as firewalls as well as operating systems. I see the full adoption of the Common Criteria evaluation scheme by all of its participating member countries as the biggest issue facing all those planning to perform a security evaluation.

Workshop Report
on
The Role of Optical Systems and Devices for Security

Panel Chair

Terry Mayfield
Institute For Defense Analyses

This panel will feature attendees of an invited workshop sponsored by The Defense Advanced Research Agency, The National Science Foundation, and The U.S. Air Force. The workshop was held at the Institute For Defense Analyses on February 26-28, 1996. The intent of this panel is to broaden the awareness of the security issues and potential for solutions using optical technology as they were discussed during the workshop.

The successful development of optoelectronic processors for security, verification, and anti-counterfeiting will impact many important government and civil sector enterprises. Because of this field's technical promise, research and development has been intensifying in many academic, government, and industrial laboratories. The goals of this workshop were to discuss various security and anti-counterfeiting topics within the context of optical technology, including major long and short term goals of the optics field and to provide a strategy for addressing R&D in support of the security needs of government and industrial users of this technology.

The report of this workshop will be published. By providing the findings of workshop, this report seeks to fill an information gap on how research in the field of optical systems and devices for security and anticounterfeiting could be maximized to the benefit of Government and industry users.

The panel reporting on this workshop will address security and vulnerabilities in all-optical networks, discuss the use of optics for information encoding, introduce some of the variety of applications that might take advantage of optical technology, and provide a summary of the workshop findings with respect to a research strategy.

The Panelists are:

Muriel Medard, MIT LIncoln Laboratory
Jeff Ingles, National Security Agency
Mark Krawczewicz, National Security Agency
Bahram Javidi, University of Connecticut

NISSC - OCT 96

Workshop Report
on
The Role of Optical Systems and Devices for Security

Panel Chair: Terry Mayfield, Institute For Defense Analyses

Panelists:

Jeff Ingle, National Security Agency
Muriel Medard, MIT Lincoln Laboratory
Mark Krawczewicz, National Security Agency
Bahram Javidi, University of Connecticut

maytield@ida.org
(703) 845-6602

IDA

Slide 1 of 6

Workshop Report
on
The Role of Optical Systems and Devices for Security

Location/Dates: Institute For Defense Analyses / 26-28 Feb 1996

Sponsors: NSF, DARPA, USAF

Chair: Prof. Bahram Javidi

Participants: 30 Leading Experts from Industry, Academia, and Government

Motivation:

Exploit Window of Opportunity to Address Security Issues & Possibilities in Emerging Optical Systems and Devices Research

Objectives:

Determine State of Research in U.S.
Identify Major R&D Goals
Propose Strategies for Achieving Goals

IDA

Workshop Report
on
The Role of Optical Systems and Devices for Security

WORKING GROUPS

Security, Including Cryptography, in Optical Networks

Applications of Optical Processing Techniques to Security Systems

Systems, Devices, and Components

Algorithms, Including Pattern Recognition and Neural Networks

Strategies to Provide R&D Funding by Government & Industry

IDA

Workshop Report
on
The Role of Optical Systems and Devices for Security

PROPOSED TECHNICAL AREAS FOR FURTHER RESEARCH

Secure, Optoelectronic and All-Optical Network Architectures

Optical Information Security Systems

High-Performance Image Information Processing

High-Performance Optical Devices and Components

IDA

Workshop Report
on
The Role of Optical Systems and Devices for Security

STRATEGIES FOR ACHIEVING GOALS

Encouraging Cross-Disciplinary and Joint Research among Researchers

- **Algorithmic**
- **Systems Integration**
- **Materials & Devices**

Identifying Industrial/Commercial Applications of Optoelectronics to Encourage R&D Funding

Focusing on How Optics can Complement Existing Systems in Supplying Security Applications

IDA

Workshop Report
on
The Role of Optical Systems and Devices for Security

PANEL PRESENTATIONS

Muriel Medard: "Security Issues for All-Optical Networks"

Jeff Ingle: "Security for All-Optical Networks"

Mark Krawczewicz: "Compact Fingerprint Scanner Techniques"

Bahram Javidi: "Optical Processing Systems for Encryption, Security Verification, and Anticounterfeiting"

IDA

SECURITY ISSUES FOR ALL-OPTICAL NETWORKS

Muriel Medard
Massachusetts Institute of Technology
Lincoln Laboratory

In Response to DoD and commercial demand for networks with increased bandwidth and extensibility, there have been many recent research efforts pursuing the development of all-optical networks. All-optical networks promise THz bandwidth, scalability, extensibility, and interoperability with legacy systems, but possess potential, as yet unstudied, security vulnerabilities. The All-Optical Network Consortium testbed we are currently building, The ONTC Testbed, the NONTC Testbed, as well as IBM's commercial RAINBOW network are examples of advanced, high-performance optical wavelength-division multiplexed (WDM) networks. The imminent deployment and use of these types of networks by a combination of DoD and Commercial users calls for a near-term study of the potential vulnerabilities, and their countermeasures. Certain of the vulnerabilities of all-optical networks are also expected to be present in electro-optical networks, and many of the countermeasures developed for all-optical networks would be directly applicable to the electro-optic counterpart, particularly when specific component vulnerabilities are concerned.

We propose research be initiated to investigate methods of increasing the security of all-optical networks against service denial, eavesdropping, traffic analysis, and unauthorized access at a level equal to or greater than in the current generation of electro-optic networks. Such research would focus on all aspects (or Network Levels) of the network, including components, subsystems, network protocols, network management, network monitoring, etc.

In our view, it is important to create a thorough theoretical understanding of the security of all-optical networks, and to understand the salient hardware characteristics that effect network security. Currently operating all-optical network testbeds, existing engineering implementations of fast networked communications, and both DoD and general societal pressure for technological solutions to privacy and other security concerns combine as an effective motivator to act now.

This work was sponsored by the Defense Advanced Research Projects Agency.

SECURITY FOR ALL-OPTICAL NETWORKS

Jeff Ingle
R22
National Security Agency
9800 Savage Rd., Suite 6516
Ft. Meade, MD 20755-6516
jtingle@alpha.ncsc.mil
301-688-0291

Scott McNown
R22
National Security Agency
9800 Savage Rd., Suite 6516
Ft. Meade, MD 20755-6516
smcnow@alpha.ncsc.mil
301-688-0291

Due to the drive for ever-increasing bandwidth, transparent, all-optical networks are beginning to emerge as a future network technology. Since the architecture and technologies are just starting to become a reality, now is the time to include security and survivability. Security and survivability include employing traditional security services and security-aware network management, and also incorporating survivable network elements in the infrastructure to provide defensive information warfare capabilities. To include security and survivability, network architecture studies are needed to understand vulnerabilities and incorporate countermeasures, and to incorporate security features in network management. As optical technologies are used to create the components for all-optical networks, survivability can be incorporated and security components like optical encryptors can be developed. Different types of transparant all-optical networks will be discussed with some security implications for each included.

NISSC - Oct 96

Workshop Report on The Role of Optical Systems and Devices for Security

Security for All-Optical Networks

Jeff Ingle
R22
National Security Agency
9800 Savage Rd., Suite 6516
Ft. Meade, MD 20755-6516

jtingle@alpha.ncsc.mil
301-688-0291 ph
301-688-0289 fax

Scott McNown
R22
National Security Agency
9800 Savage Rd., Suite 6516
Ft. Meade, MD 20755-6516

srmcnow@alpha.ncsc.mil
301-688-0291 ph
301-688-0289 fax

Vision of Emerging All-Optical Networks

Motivation

- Drive for more bandwidth - will push aggregate rates of ATM/SONET past highest defined OC-192 (9.6 Gbps)
- Need to protect information - both information security (INFOSEC) and survivability
- INFOSEC (security mechanisms) and Survivability (counter vulnerabilities) share solutions
- Opportunities to incorporate INFOSEC and survivability into emerging networks and standards

Vision of Emerging All-Optical Networks

Emerging All-Optical Networks

- Near-term technology
  - Time Division Multiplexing (TDM) combined with Wavelength Division Multiplexing (WDM)
  - Probably multiple OC-192 (9.6 Gbps TDM) channels, on multiple wavelengths (WDM)
- Circuit-switched all-optical networks
- Packet-switched all-optical networks
- Longer-term technologies
  - Solitons, CDMA, Quantum communications, Wideband coherent comm.

Research approach for security and survivability

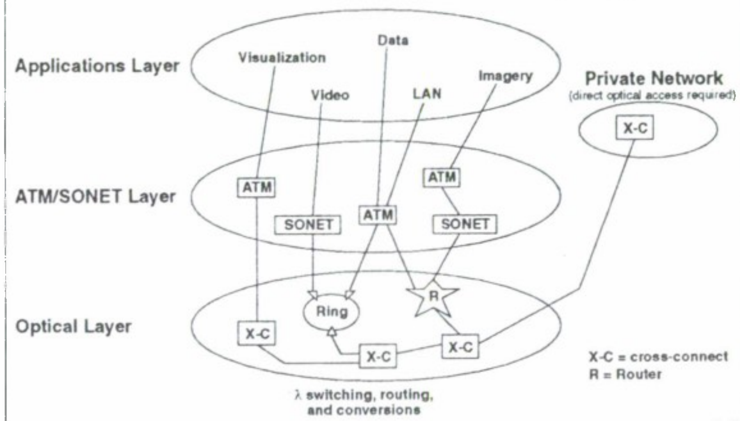
Network Architecture Study

- Vulnerabilities and Countermeasures
- Security in Network Management
- Hooks for other security and survivability

Research to develop devices and components

- To incorporate countermeasures in vulnerable components
- To provide security mechanisms like confidentiality (encryptors)

Circuit-switched all-optical network topology



Circuit-switched optical networks

Network Architecture Study

- Architecture
  - topology, network node composition, service provisioning and signaling
  - security implications of "just-in-time" signaling for minimal latency
- Network Control and Management (NC&M)
  - fault detection and localization, configuration management, quality of service (QoS) management, security management, resource allocation
- Authentication
  - of end users, signaling, QoS negotiation
  - for access control, accounting and billing
- Security service negotiation capability
  - level of security, type of encryption and key exchange algorithm, authentication protocol, data integrity, etc. (possible model in IPv6)

Research to develop devices and components

- Survivability of common optical network components
  - Optical multiplexers, optical routers, optical amplifiers
  - Reduce vulnerability to jamming, reduce crosstalk, organize subsystems within component for best resistance
  - Develop - comprehensive set of design rules, methods to counter attacks, robust devices
- Confidentiality and key management
  - Use symmetric encryption algorithms (e.g. DES) for high speed encryption, public key cryptography to distribute keys (although slow)
  - Extend SONET encryptor model to WDM environment
  - Need for optical encryptors is niche market - DoD, DOE, NASA supercomputing facilities
- Network Security Managers

Packet-switched optical networks

Difficult challenge - emerging architectural possibilities

Network Architecture Study

- Follow similar approach as for circuit-switched optical networks
  - Authenticated signaling, flexible security negotiation mechanisms, security fields in signaling for crypto sync/resync - especially when no initial end-to-end connectivity

Research to develop devices and components

- Counter vulnerabilities in network components like switches or routers
- Optical packet encryptor
  - Word-based - one-dimensional string of bytes
  - Page-based - two-dimensional array of bytes
  - Need packet identifier, key generator (KG), optical delay, optical XOR

Longer-term technologies

Soliton transmission technology

- near term implementation in intercontinental submarine links
- could emerge as long term network technology
- confidentiality - mux parallel encryptors or high-speed cryptographic algorithm in technology like fiber loop logic

Code Division Multiple Access (CDMA)

- optical spread spectrum techniques
- privacy system, limited in distance and networking
- may be possible to use very fast cryptographic algorithm and technology to implement for high security

Quantum Communications

- Quantum Cryptography
  - high security (not based on public key techniques like factoring)
  - limited bandwidth and distance - cannot network
  - might be used for key distribution
- may reduce threat of covert channels
- may not be feasible in network situation

Wideband coherent communications

- may reduce threat of covert channels
- may not be feasible in network situation

OPTICAL PROCESSING SYSTEMS FOR ENCRYPTION, SECURITY VERIFICATION, AND ANTICOUNTERFEITING

Bahram Javidi

University of Connecticut, U-157
Electrical and Systems Engineering Department
260 Glenbrook Road
Storrs, Connecticut 06269-2157
Tel. 203-486-2867 Fax 203-486-1273 email: bahram@eng2.uconn.edu

The recent development of commercially available, low-cost optoelectronic devices, components, and systems and their increased technical performance suggest that optical processing systems and devices have significant potential for encryption, security, verification, and anticounterfeiting applications.These systems can combat fraud which is a serious problem facing many banks, businesses, and consumers. Counterfeit parts such as computer chips, machine tools, etc. are arriving on our shores in great numbers. With the rapid advances in computers, CCD technology, image processing hardware and software, printers, scanners, and copiers, it is simple to reproduce pictures, logos, symbols, money bills, or patterns. Presently, credit cards and passports use holograms for security which are inspected by human eye and can be easily reproduced.

We present a number of new all optical encoding methods that can perform high speed encryption of high volume of data such as images in parallel into complex phase/ amplitude patterns .[1] These techniques can be used for encryption, anti-counterfeiting, security verification of credit cards, passports, and other IDs so that they cannot easily be reproduced. Complex phase/amplitude patterns that cannot be seen and cannot be copied by an intensity sensitive detector such as a CCD camera are employed to encode the data.

The decryption/reconstruction can be performed in parallel and is both fault tolerant and robust to noise. These techniques allow encoding of information in a way which is difficult to decode if one does not know a ``key" but very easy if one knows that key.

One method is to permanently and irretrievably bond a 2-D phase mask to a primary identification amplitude pattern such as a fingerprint, a picture of a face, or a signature. Both the phase mask and the primary pattern are identifiable in an optical processor or correlator.[2] The phase portion of the pattern consists of a 2-D phase mask with large dimensions to make it difficult to determine the contents of the mask. With the high resolution of the commercially available optical films and materials, $M(x,y)$ can be of the order of a million pixels, and yet the mask size will be only a few millimeter square.

The verification system that reads the card could be one of several coherent optical processors architectures. This system provides the flexibility of allowing a "code of the day", or whatever time interval is appropriate, by inserting a card containing another authorized reference phase mask in the input. For even more security, the primary pattern could itself be phase encoded. This would have the effect that the combined pattern would be completely invisible to the eye or to any other detector using conventional light sources. This technique would have an additional security value, in that anyone wanting to counterfeit the card, would not even be able to easily determine what type of a primary pattern they would have to produce on the card.

1. P. Refregier and B. Javidi, Optical Image Encryption using Input and Fourier Plane Random Phase Encoding," Journal of Optics Letters, vol. 20, pp. 767-769, April 1, 1995.

2. B. Javidi, and J. L. Horner, "Optical Pattern Recognition for Validation and Security Verification," Journal of Optical Engineering, vol. 33, no. 6, June 1994.

NISSC - Oct 96

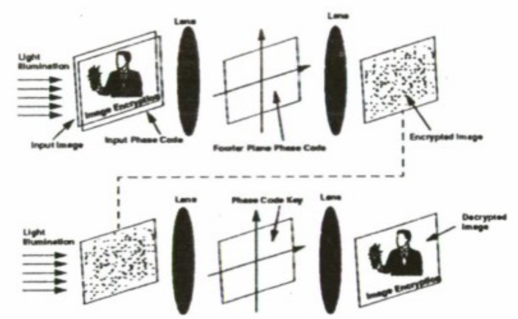
Workshop Report on
The Role of Optical Systems and Devices for Security

OPTICAL PROCESSING SYSTEMS FOR ENCRYPTION, SECURITY VERIFICATION, AND ANTICOUNTERFEITING

Bahram Javidi
University of Connecticut, U-157
Electrical and Systems Engineering Department
260 Glenbrook Road
Storrs, CT 06260-2157

bahram@eng2.uconn.edu
Phone - 203-486-2867
Fax - 203-486-1273

HIGH SPEED PARALLEL IMAGE ENCRYPTION

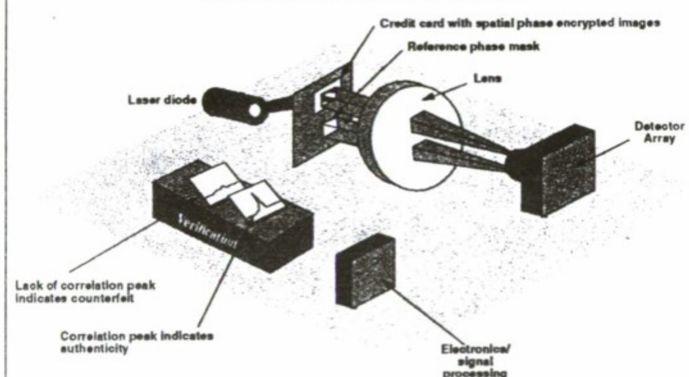


P. Refregier and B. Javidi, "Optical Image Encryption Using Input and Fourier Plane Random Phase Encoding," Journal of Optical Letters, Vol. 20, pp. 767-769, April 1, 1995.

SPATIAL PHASE ENCRYPTED CREDIT CARD

B. Javidi and J.L. Horner, "Optical Pattern Recognition for Validation and Security Verification," Optical Engineering, Vol. 33, no. 6, June 1994.

OPTO-ELECTRONIC PATTERN RECOGNITION SYSTEM FOR VERIFYING SPATIAL PHASE ENCRYPTED CARDS



B. Javidi and J.L. Horner, "Optical Pattern Recognition for Validation and Security Verification," Optical Engineering, Vol. 33, no. 6, June 1994.

INFORMATION SECURITY CHALLENGES IN THE FINANCIAL SERVICES INDUSTRY

C. Thomas Cook
Executive Vice President
Banc One Services Corporation
Columbus, Ohio

Until recently the financial services industry has been supported by closed business systems. Consequently, information security activities have adhered to traditional venues which emphasize protecting information assets from internal threats. However, as the landscape of the financial services industry changes, emerging technologies are being deployed to support the shift in the business paradigm. As these technologies are implemented, enterprises are faced with a new set of information security threats.

The financial services industry, like other industries, is using emerging technologies to redefine the way business is conducted and exploit new distribution channels. Business transactions that were once paper based are now conducted electronically. Customers now complete financial transactions by connecting to our business systems via the telephone or the PC. As the migration of the our industry to open systems, distributed computing and electronic commerce continues to accelerate, so does the proliferation of new vulnerability points at the point of information access.

All of these factors combined are resulting in a new focus on information security by legal, regulatory and internal audit areas. Why? Because these factors will cause enterprises to contend with a new set of threats that hold the potential to harm systems, information assets and the business by increasing susceptibility to risk, including:

- Data manipulation
- Disclosure of sensitive data
- Destruction of information assets
- Disruption of business

These threats put information security risk management at a crossroads, which might ultimately result in the redefinition of the function within the next five years (Gartner). In the meantime, this new age of information security requires the understanding of senior managers, well-defined business processes and technology to measure and monitor risks.

Mr. Cook will discuss how the rapid deployment of new technologies changes the information security profile in the financial services industry and the requirements of information security strategies, base-line controls and products that are necessary to mitigate exposure.

INFORMATION SYSTEMS AUDITING REQUIREMENTS

John W. Lainhart IV
Inspector General
U.S. House of Representatives
485 Ford House Office Building
Washington, D.C. 20515-9990

Not a new direction or challenge for information systems security, but a direction often not pursued and a challenge often not addressed, information systems auditing is critically required in today's information systems intensive environment. It is required to ensure that our mission critical or lifeblood systems are designed and continue to be maintained with confidentiality, integrity, and availability foremost in mind. In addition, information systems auditing is required by professional auditing standards when information systems are involved in the area being audited. To assist in this effort, a new set of standards, *CobiT* (Control Objectives for Information and Related Technology) was recently issued which contains both information technology (IT) control objectives, for management and users, and information systems audit guidelines, for auditors.

Standards Relating to Audits Involving Information Systems

The American Institute of Certified Public Accountants (AICPA) in several *Statements on Auditing Standards* (SASs), Institute of Internal Auditors Association (IIA) in its *Standards for the Professional Practice of Internal Auditing*, Information Systems Audit and Control Association (ISACA) in its *General Standards for Information Systems Auditors* and *Statements on Information Systems Auditing Standards*, and U.S. General Accounting Office (GAO) in its *Government Auditing Standards* and *Title 2*, *Accounting*, have all taken essentially the same position concerning audits involving information systems. The bottom line is that when an information system is an important and integral part of the operations being audited, the audit should include an appropriate examination of the system to provide reasonable assurance that the information produced by the system is valid and reliable (relevant, accurate, and complete, in light of its intended use).

Specifically, GAO's *Government Auditing Standards* states that "auditors should obtain sufficient, competent, and relevant evidence that computer processed data are valid and reliable when those data are significant to the auditors' findings." The *Government Auditing Standards* goes on to state that "when the reliability of a computer-based system is the primary objective of the audit, the auditors should conduct a review of the system's general and application controls." Furthermore, in its *Appendix III*, *Accounting System Standards*, *Chapter 4*, *Accounting System Development and Modification*, of *Title 2*, GAO states that Offices of Inspectors General (OIGs) are an important factor contributing to successful accounting and financial management system development and modification efforts. GAO indicates that while normally not a member of the project team, auditor involvement is needed in reviewing and evaluating these development and modification efforts.

CobiT -- Control Objectives for Information and Related Technology

In order to facilitate this review of information systems, ISACA recently issued *CobiT*. It was developed as a generally applicable and accepted international standard for good practices for IT controls. *CobiT* is based on ISACA's existing *Control Objectives*, enhanced with existing and emerging international technical, professional, regulatory, and industry-specific standards. It was written for three specific audiences -- management, users, and auditors. By using this document, management will be able to review the organization's information systems to make IT investment decisions, balance risks and controls, and benchmark its existing and future IT environments. Users will be able to obtain assurance on the security and control of products they acquire (internally or externally). Finally, auditors will be able to substantiate internal control opinions and identify needed minimum controls for management.

CobiT identifies 4 domains with 32 IT processes which form the *Framework* for from 5 to 25 detailed *Control Objectives*. The first domain, planning and organization, covers strategy and tactics and concerns the identification of the way IT can best contribute to the achievement of business objectives. It also emphasizes that a proper organization, as well as technological infrastructure, must be in place. The second domain, acquisition and implementation, recognizes that to realize the IT strategy, IT solutions need to be identified, developed, or acquired as well as implemented and integrated into the business process. It also addresses changes to and maintenance of existing systems. The third domain, delivery and support, is concerned with the actual delivery of required services, which range from traditional operations over security and continuity aspects to training. This domain also includes the actual processing of data by application systems. The final domain, monitoring, recognizes that all IT processes need to be regularly assessed over time for their quality and compliance with control requirements.

In addition to the domains, processes, and control objectives which are used by management, users, and auditors, *CobiT* provides detailed *Audit Guidelines* for auditors to follow in performing information systems audits -- thereby, meeting their information systems auditing requirements! Thus, the *Audit Guidelines* provide a complementary tool to enable the easy application of the *Framework* and *Control Objectives* within audit activities. *CobiT* states that the objectives of auditing are to: (1) provide management with reasonable assurance that control objectives are being met; (2) where there are significant control weaknesses, to substantiate the resulting risks; and (3) advise management on corrective actions needed (ones needed at a minimum, and ones that are cost beneficial). *CobiT* goes on to state that information systems are audited by: (1) **obtaining an understanding** of business requirements related risks, and relevant control measures; (2) **evaluating the appropriateness** of stated controls; (3) **assessing compliance** by testing whether the stated controls are working as prescribed, consistently and continuously; and (4) **substantiating the risk** of control objectives not being met by using analytical techniques and/or consulting alternative sources.

Conclusion

Clearly, information systems auditing is mandated by an abundance of specific professional standards -- from both public and private accounting and auditing organizations. But even more important is the need of our organizations for increased quality, decreased delivery time, and continuous service level improvements. All these aspects must be achieved within tighter cost constraints, with fewer resources. At the same time, assets of the organization must be adequately safeguarded, and for many organizations, information and the technology that supports it represent the organization's most valuable assets. Thus, it just makes good sense to aggressively audit information systems -- both those that are operational (general and application systems) and those that are under development or modification.

INFORMATION SYSTEMS AUDITING REQUIREMENTS

INFORMATION SYSTEMS AUDITING REQUIREMENTS

STANDARDS RELATING TO AUDITS INVOLVING INFORMATION SYSTEMS

COBIT (CONTROL OBJECTIVES FOR INFORMATION AND RELATED TECHNOLOGY)

- ***STANDARDS - INFORMATION TECHNOLOGY CONTROL OBJECTIVES***
- ***STANDARDS - INFORMATION SYSTEMS AUDIT GUIDELINES***

STANDARDS RELATING TO AUDITS INVOLVING INFORMATION SYSTEMS

AICPA - SAS NOs. 48, 55, 70 & 78

IIA - *STANDARDS FOR THE PROFESSIONAL PRACTICE OF INTERNAL AUDITING*

ISACA - *GENERAL STANDARDS FOR INFORMATION SYSTEMS AUDITORS*

- *STATEMENTS ON INFORMATION SYSTEMS AUDITING STANDARDS*

STANDARDS RELATING TO AUDITS INVOLVING INFORMATION SYSTEMS

GENERAL ACCOUNTING OFFICE

* ***GOVERNMENT AUDITING STANDARDS***

* ***TITLE 2, ACCOUNTING, APPENDIX III, ACCOUNTING SYSTEM STANDARDS, CHAPTER 4, ACCOUNTING SYSTEM DEVELOPMENT AND MODIFICATION***

CobiT: FRAMEWORK

Audience – Management:

To Make IT Investment Decisions, Balance Risks and Controls, and Benchmark IT Environments

Audience – Users:

To Obtain Assurance on Security and Control of Products and Services Acquired

Audience – Auditors:

For Management, Substantiate Internal Control Opinions and Identify Minimum Controls Necessary

CobiT: CONTROL OBJECTIVES

The DOMAINS

* *Planning & Organization*
* *Acquisition & Implementation*
* *Delivery & Support*
* *Monitoring*

CobiT: AUDIT GUIDELINES

The objectives of auditing are to:

- provide management with reasonable assurance that control objectives are being met;

- where there are significant control weaknesses, to substantiate the resulting risks; and

- advise management on corrective actions.

CobiT: AUDIT GUIDELINES

The process is audited by:

***Obtaining an understanding** of business requirements related risks, and relevant control measures*

***Evaluating the appropriateness** of stated controls*

***Assessing compliance** by testing whether the stated controls are working as prescribed, consistently and continuously*

***Substantiating the risk** of the control objectives not being met by using analytical techniques and/or consulting alternative sources.*

Willis H. Ware
26 July 1996

Since the technical aspects of computer security were first highlighted during the 1970s, the research community of the United States has examined the question of reliable software security safeguards in operating systems and major other sysstem components. Given the defense orientation of security from its beginnings in the late 60s, such work has largely focused on the defense context -- its perceived threat, operational environment, security requirements, operational policies. The 25-year effort has culminated in a series of "criteria documents" from several countries, but the most comprehensive of which is the currently emerging Common Criteria, a joint effort of all countries.

Is the philosophy inherent in a criteria approach sufficiently comprehensive for the security needs of business, industry and civil government? Are there security dimensions that a criteria approach cannot, or is not likely, to respond to? Is the Common Criteria adequately general to encompass the varied needs of non-defense systems and installations? What are the specific security needs perceived in the non-defense environment, and how to they differ from those in defense? Are there dimensions of the non-defense threat to which a criteria approach might be provide inadequate or awkward safeguards?

This panel will discuss and explore such a range of issues, with emphasis on highlighting specialized security needs from civil government and the private sector. It is hoped that new suggestions for research directions might be identified, that hitherto unrecognized security requirements will surface, that differences between the operational environments in defense and non-defense will emerge, and that provisional answers to the set of questions above might develop.

Panel

The Next Generation of Cybercriminals

Moderator
Mark Gembicki
Executive Vice President
WarRoom Research, LLC
(410) 437 - 1106

Panelists

Jim Christy
Chief, Computer Crime Investigations & Information Warfare
Air Force Office of Special Investigations

Bill Perez
Unit Chief, Financial Institution Fraud Unit (& Computer Crime)
Federal Bureau of Investigation

Doug Waller
National Security Correspondent
Time Magazine

Summary

What is the present state of cybercrime? Why do "cybercartels" pose such a unique threat? What role does legal competitive intelligence play? How safe is our National Information Infrastructure? Is the commercial business environment the real target for information warriors? A diverse panel of experts present answers to these and other intriguing questions in attempt to foster a better understanding of The Next Generation of Cybercriminals.

Examples of criminal organizations will be presented, which range from traditional organized crime elements to the individual "cyberterrorist." Findings from the Security in Cyberspace hearings will be presented. Proposed changes on broadening U.S. Government investigations of cybercriminals and increasing the reporting of criminal activities by commercial organizations will be discussed. Also, methods will be outlined to better safeguard organizational assets, such as knowledge and technology, from the cybercriminals and intelligence gatherers.

Throughout the panel discussion, members will address audience questions in several areas relevant to the cybercrime topic. Specifically, the areas of white-collar crime, information systems security, operations security, information warfare, competitive intelligence and national security are welcome.

☆ U.S. GOVERNMENT PRINTING OFFICE: 1996 -- 415 - 261